

A Contour Grouping Algorithm for 3D Reconstruction of Biological Cells

by

Tony Kin Shun Leung

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Civil Engineering

Waterloo, Ontario, Canada, 2009

©Tony Kin Shun Leung 2009

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Tony Kin Shun Leung

Abstract

Advances in computational modelling offer unprecedented potential for obtaining insights into the mechanics of cell-cell interactions. With the aid of such models, cell-level phenomena such as cell sorting and tissue self-organization are now being understood in terms of forces generated by specific sub-cellular structural components. Three-dimensional systems can behave differently from two-dimensional ones and since models cannot be validated without corresponding data, it is crucial to build accurate three-dimensional models of real cell aggregates. The lack of automated methods to determine which cell outlines in successive images of a confocal stack or time-lapse image set belong to the same cell is an important unsolved problem in the reconstruction process. This thesis addresses this problem through a contour grouping algorithm (CGA) designed to lead to unsupervised three-dimensional reconstructions of biological cells.

The CGA associates contours obtained from fluorescently-labeled cell membranes in individual confocal slices using concepts from the fields of machine learning and combinatorics. The feature extraction step results in a set of association metrics. The algorithm then uses a probabilistic grouping step and a greedy-cost optimization step to produce grouped sets of contours. Groupings are representative of imaged cells and are manually evaluated for accuracy.

The CGA presented here is able to produce accuracies greater than 96% when properly tuned. Parameter studies show that the algorithm is robust. That is, acceptable results are obtained under moderately varied probabilistic constraints and reasonable cost weightings. Image properties – such as slicing distance, image quality – affect the results. Sources of error are identified and enhancements based on fuzzy-logic and other optimization methods are considered. The successful grouping of cell contours, as realized here, is an important step toward the development of realistic, three-dimensional, cell-based finite element models.

Acknowledgements

I would like to express gratitude to Dr. G. Wayne Brodland for his guidance and support during the course of completing this thesis. It was only with his patience and ideas that such a work was possible. His supervision has provided many learning opportunities, of which I will continue to value greatly as I travel along the road of life.

I also appreciate the tireless help of Caleb Horst and Jim Veldhuis. Their expertise and knowledgeable opinions have undoubtedly contributed to the completion of this thesis.

Special thanks also goes to my fellow students Paul Groh, Simon Tsui, P. Graham Cranston and Justina Yang. Their support and friendship made this challenging journey both fun and exciting.

Finally, I would like thank my family for their unconditional support and love. To my mother and father; Rebecca Leung and Kenneth Leung, thank you for providing the perfect environment in which to grow, to learn, and to become all that I strive become. To my siblings; Sam Leung and Danny Leung, thank you for being the perfect role models, for being patient, and for showing me what it means to be proud of one's elder siblings. To the soon-to-be Mrs. Sam Leung; Amy Ng, you are truly like a sister.

Table of Contents

List of Figures.....	vii
List of Tables.....	x
Chapter 1: Introduction	1
Chapter 2: Background	3
2.1 Cell-Level Finite Element Analysis (FEA).....	3
2.1.1 The Three-Dimensional Model (Viens and Brodland, 2007)	3
2.1.2 Model Input Requirements	6
2.2 Imaging Modalities	6
2.2.1 Magnetic Resonance Imaging (MRI).....	6
2.2.2 Computed Tomography (CT)	8
2.2.3 Fluorescent Confocal Microscopy (FCM).....	10
2.2.4 Comparison of MRI, CT and FCM.....	12
2.2.5 Three-Dimensional Reconstructions	13
2.3 Data Clustering and Optimization	15
Chapter 3: Contour Grouping Algorithm.....	19
3.1 Source Data	19
3.2 Image Properties and Metrics	21
3.2.1 Geometric Properties: 2D Region Shape	22
3.2.2 Geometric Properties: Region Area.....	25
3.2.3 Coordinate Properties: Region Position and Neighbours.....	27
3.2.4 Coordinate Properties: Region Offset	29
3.2.5 Coordinate Properties: Region Topology (Same Slice Neighbourhood)	34
3.2.6 Metrics Summary	38
3.3 Contour Grouping Algorithm.....	39
3.3.1 Data Preparation	39
3.3.2 Probabilistic Iterative Associations.....	42
3.3.3 Greedy Cost Association.....	55
3.3.4 Association Visualization	58
Chapter 4: Algorithm Details	60
4.1 Parameter Choice	60

4.1.1	Gaussian Parameters	60
4.1.2	Probabilistic Thresholds	61
4.1.3	Cost Function Parameters.....	63
4.2	Error Types.....	67
4.2.1	Singe Cell Errors	67
4.2.2	Adjacent Cell Errors.....	69
4.2.3	Unassigned Contours	73
4.3	Evaluation Measures.....	73
Chapter 5: Algorithm Performance		75
5.1	Synthetic Data	75
5.1.1	Curved Cell Sheet	76
5.1.2	Cell Aggregate	78
5.1.3	Discussion.....	80
5.2	Parameter Changes.....	81
5.2.1	Curved Cell Sheet	82
5.2.2	Cell Aggregate	84
5.2.3	Observations	85
5.3	Slicing Distance Changes.....	86
5.3.1	Results.....	88
5.3.2	Evaluation.....	90
5.4	Real Image Data	91
5.4.1	Fly-wing Z- Series (Axelrod).....	92
5.4.2	Fly-wing Z- Series (Axelrod) – Modified Parameter Value	99
5.4.3	Drosophila Wound Time Series (Antonio)	100
5.4.4	Observations	103
5.5	Summary	104
Chapter 6: Conclusions and Future Work.....		105
6.1	Image Properties and Quality	105
6.2	Metrics and Parameters.....	105
6.3	Algorithm Performance and Efficiency	106
References.....		107

List of Figures

Figure 1.1: Sample confocal slices	1
Figure 1.2: Synthetic cell aggregate	2
Figure 2.1: FEA cell representation (Yang, 2008)	4
Figure 2.2: Cellular components modelled (Viens and Brodland, 2007).....	4
Figure 2.3: Calculation of nodal forces (Viens and Brodland, 2007)	5
Figure 2.4: Orthogonal dashpot cell-model for single direction (Yang, 2008)	6
Figure 2.5: MRI image of a human skull (“A to Z of tinnitus”).....	7
Figure 2.6: CT image of human thoracic cavity (Klingenbeck-Regn et al., 1999)	8
Figure 2.7: CT concept (Ramm and Katsevich, 1996)	9
Figure 2.8: FCM image of laser-induced wound in Drosophila	11
Figure 2.9: Confocal optics (Groh, 2008).....	11
Figure 2.10: Sample Delaunay surface reconstructions (Wang et al., 2006)	14
Figure 3.1: Cell aggregate.....	19
Figure 3.2: Isolated cells 3D view	20
Figure 3.3: Sample slices	20
Figure 3.4: 3D region associations.....	21
Figure 3.5: Cell shape and equivalent ellipses.....	23
Figure 3.6: Distribution of cell kappa values	24
Figure 3.7: Distribution of cell kappa values within slice	24
Figure 3.8: Slice 3D cell.....	25
Figure 3.9: Superimposed region areas.....	25
Figure 3.10: Pixel overlap	26
Figure 3.11: 4-nearest adjacent slice neighbours diagram	28
Figure 3.12: Diagram of all adjacent slice neighbours	29
Figure 3.13: Manually associated slice regions	30
Figure 3.14: Displacement vector fields	30
Figure 3.15: Superimposed vector fields.....	31
Figure 3.16: Bending visualization.....	31
Figure 3.17: Bending angle comparison	32
Figure 3.18: Bending metric Gaussian fit	33
Figure 3.19: Region topology.....	34
Figure 3.20: Grouping topology example.....	35
Figure 3.21: Direct and secondary topology	36
Figure 3.22: Topology comparison example	36
Figure 3.23: Region comparisons	37
Figure 3.24: Sample volume and image stack	40
Figure 3.25: Pixel overlap associations.....	41
Figure 3.26: Bending associations	41
Figure 3.27: Triplet description	42

Figure 3.28: Z-extension diagram	43
Figure 3.29: Conditional triplet diagram.....	44
Figure 3.30: Topology check diagram	45
Figure 3.31: X-Y extension diagram	46
Figure 3.32: Sample cell sheet	48
Figure 3.33: Probabilistic association result	48
Figure 3.34: Curved cell-sheet	50
Figure 3.35: Metric conversion to cost	55
Figure 3.36: Type 1 optimization assignment.....	57
Figure 4.1: Normalized Gaussian Distribution	61
Figure 4.2: Gaussian fit regions.....	61
Figure 4.3: Overlap and topology cost.....	64
Figure 4.4: Bending and topology cost	65
Figure 4.5: Overlap, bending and topology cost.....	65
Figure 4.6: Penalty cost.....	66
Figure 4.7: Typical Type 1 error	67
Figure 4.8: Slice-by-slice Type 1 error.....	68
Figure 4.9: Slice-by-slice Type 1 error resolution.....	68
Figure 4.10: Series type 1 error.....	69
Figure 4.11: Diagram of adjacent cells.....	69
Figure 4.12: Typical Type 2 error	70
Figure 4.13: Case A Type 2 error	70
Figure 4.14: Case B Type 2 error	71
Figure 4.15: Series Type 2 error	71
Figure 4.16: Typical Type 3 error	72
Figure 4.17: Slice-by-slice Type 3 error.....	72
Figure 4.18: Unassigned contours.....	73
Figure 5.1: 100-cell curved sheet.....	76
Figure 5.2: Cell orientations.....	77
Figure 5.3: Curved cell sheet results.....	77
Figure 5.4: 50-cell aggregate.....	79
Figure 5.5: Cell aggregate result	79
Figure 5.6: Slicing differences	87
Figure 5.7: Hybrid error	89
Figure 5.8: Hybrid error resolution	89
Figure 5.9: Real data analysis steps	91
Figure 5.10: Image artifacts	94
Figure 5.11: CGA errors due to artifacts	95
Figure 5.12: Fly-wing edge tracings	96
Figure 5.13: CGA results.....	97
Figure 5.14: CGA results and error resolution	98
Figure 5.15: Fly-wing parameter change results.....	99

Figure 5.16: Drosophila wound results.....	100
Figure 5.17: Drosophila wound results -- first 3 time steps	103
Figure 5.18: Drosophila wound results errors -- first 3 time steps	103

List of Tables

Table 2.1: Comparison of MRI, CT and FCM	12
Table 3.1: Image information.....	21
Table 3.2: Derived and measured properties	22
Table 3.3: Example pixel overlap metric.....	27
Table 3.4: Bending Gaussian fit properties	33
Table 3.5: Topology comparison (Region A and B)	37
Table 3.6: Topology comparison (Region C and D).....	38
Table 3.7: Metrics list.....	39
Table 3.8: Summary of approach.....	39
Table 3.9: Summary of probabilistic association algorithm.....	46
Table 3.10: List of adjustable parameters.....	47
Table 3.11: Step by step probabilistic association depiction.....	49
Table 3.12: Algorithm iterations -- curved cell-sheet	50
Table 3.13: Algorithm iterations - curved cell-sheet (cont.).....	51
Table 3.14: Algorithm iterations - curved cell-sheet (cont.).....	52
Table 3.15: Algorithm iterations - curved cell-sheet (cont.).....	53
Table 3.16: Algorithm iterations - curved cell-sheet (cont.).....	54
Table 3.17: Cost notation.....	56
Table 3.18: Visualization of iterations (region association).....	58
Table 3.19: Final result visualization (region association)	59
Table 4.1: Error details.....	67
Table 4.2: Evaluation measure definition	74
Table 5.1: Initial Parameter Values.....	75
Table 5.2: Cell orientations	76
Table 5.3: Curved cell sheet algorithm statistics	78
Table 5.4: Curved cell sheet evaluations	78
Table 5.6: Cell aggregate algorithm statistics	80
Table 5.7: Cell aggregate evaluations	80
Table 5.8: Parameter Affects	81
Table 5.9: Parameter change cases	81
Table 5.10: Parameter change results -- cell sheet.....	82
Table 5.11: Parameter change evaluations -- cell sheet.....	83
Table 5.12: Parameter change results -- cell aggregate	84
Table 5.13: Parameter change evaluations -- cell aggregate.....	85
Table 5.14: Slicing parameters.....	87
Table 5.15: Slicing results.....	88
Table 5.16: Slicing evaluations.....	90
Table 5.17: Fly-wing results overview.....	92
Table 5.18: Fly-wing algorithm statistics.....	98

Table 5.19: Fly-wing triplet statistics.....	98
Table 5.20: Fly-wing errors.....	98
Table 5.21: Fly-wing evaluations.....	99
Table 5.22: Fly-wing parameter change.....	99
Table 5.23: Drosophila wound overview.....	100
Table 5. 24: Drosophila wound image artifacts and edge tracings.....	102

Chapter 1: Introduction

Finite element models hold remarkable promise for determining forces at work in biological cells. The power of these models is that they can allow forces too small to measure directly in the cellular environment to be estimated. However, such cell-level computational models are of little practical use without data from real tissues.

Recent three-dimensional (3D) finite element models have shown that 3D systems can behave much differently than two-dimensional (2D) systems (Hutson et al., 2008). Thus, 3D reconstructions of real cells are needed to build realistic models. Fundamentally, 3D reconstruction is a four-part task: 1) image acquisition, 2) outlining of individual cells to obtain contours, 3) contour association across slices and 4) finite-element mesh generation. An important unsolved problem in this list of steps is that of contour association.

As an exercise, the reader is asked to consider Figure 1.1a through 1.1c. Each sample slice is generated from the synthetic cell mass shown in Figure 1.2, and represents a subset of potential 3D data. Given these slices, is it possible to visually obtain contour associations? In particular, note the contours labeled '1' and '2' respectively. Cells in the aggregate are "appearing" or "disappearing" from one image to the next. Making correct contour associations is difficult to do even by hand. Automated software to make such association is the objective of this thesis.

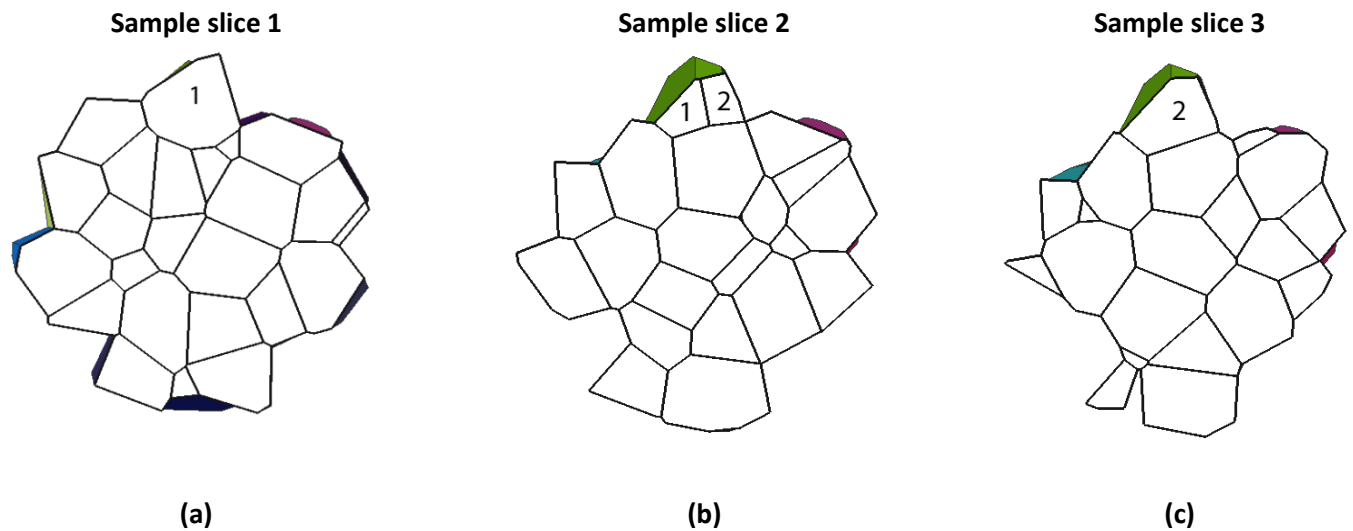


Figure 1.1: Sample confocal slices

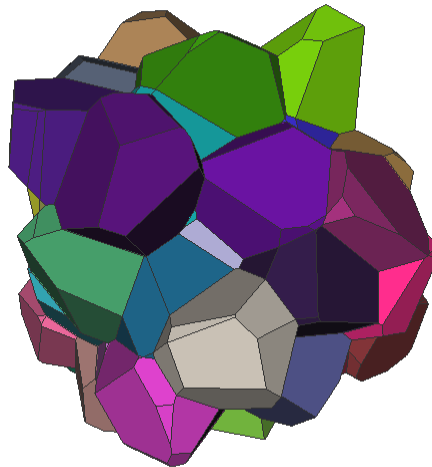


Figure 1.2: Synthetic cell aggregate

This thesis describes the development of a contour grouping algorithm (CGA) to be used in 3D reconstructions of biological cells. Images from confocal stacks were manipulated using image processing techniques and the resulting images were segmented using watershed methods. The outlines of individual cells were then extracted from these segmented images, and provide the starting point for this thesis. A set of image features (metrics) is defined for each contour and is used by the CGA. Once these contours are correctly associated across slices, subsequent generation of a 3D finite-element mesh is conceptually not difficult, but is beyond the scope of this thesis.

The successful development of a CGA is a significant step towards obtaining 3D reconstructions. Ultimately, finite element analyses will be conducted based on these reconstructions and hopefully they will lead to an improved understanding of the physical forces that drive various cell-cell rearrangements including those involved in early embryo development, cancer metastasis and tissue engineering.

Chapter 2: Background

The stated purpose of this thesis is to provide valid 3D reconstructions of biological cells for use in 3D finite element analysis (FEA). As such, it is necessary to understand the specific data requirements of these computational models. As an example, the 3D, cell-based model of Viens and Brodland (2007) is considered (Section 2.1).

Three-dimensional reconstruction methods are well known, especially in medical applications, and will be discussed in Section 2.2. A number of current imaging modalities, such as magnetic resonance imaging (MRI) and computed tomography (CT), provide volumetric data for generation of 3D reconstructions. It will be shown that the information present in MRI and CT images is fundamentally different than that obtained from fluorescent confocal microscopy (FCM). Thus the methodologies used in typical medical imaging approaches are not suitable for constructing cell-level FEA models.

Following this, Section 2.3 provides samples of machine learning concepts relevant to the CGA that is developed in this thesis. Choosing suitable contour properties is integral to the CGA, since strong metrics provide the basis for reliable association decisions.

2.1 Cell-Level Finite Element Analysis (FEA)

Continued advances in computing power have led to the development of powerful computer models for biology. Models for cell-cell signaling (Barbier De Reuille et al., 2006), morphogenesis (Longo et al., 2004; Chen and Brodland, 2007) and cell proliferation (Boman et al., 2001) are continually being developed and refined. Computational models are attractive because they allow information to be inferred about the forces generated by structures too small to measure directly.

The three-dimensional FEA (Viens and Brodland, 2007) considered for this thesis was developed as an extension to an existing two-dimensional model developed by Brodland and Chen (2000). The two-dimensional model has been extremely useful, leading to the formulation of the Differential Interfacial Tension Hypothesis (DITH), and it was subsequently extended to include the process of cell mitosis (Brodland and Veldhuis, 2002). The three-dimensional model is more powerful, and reveals that two- and three-dimensional system can behave in quite different ways (Hutson et al., 2008). Here, only the three-dimensional model will be discussed, and the purpose of the discussion is to demonstrate the kinds of mathematical objects that the reconstruction process aims to make possible.

2.1.1 The Three-Dimensional Model (Viens and Brodland, 2007)

Figure 2.1 (a) show a synthetic set of four close-packed cells. In a typical FEA formulation, a cell is defined as a collection of faces, edges and vertices. Each face constitutes the shared boundary between 2 cells, while each edge is the shared boundary between 3 faces. Accordingly, each vertex is the shared boundary between 4 edges (Yang, 2008).

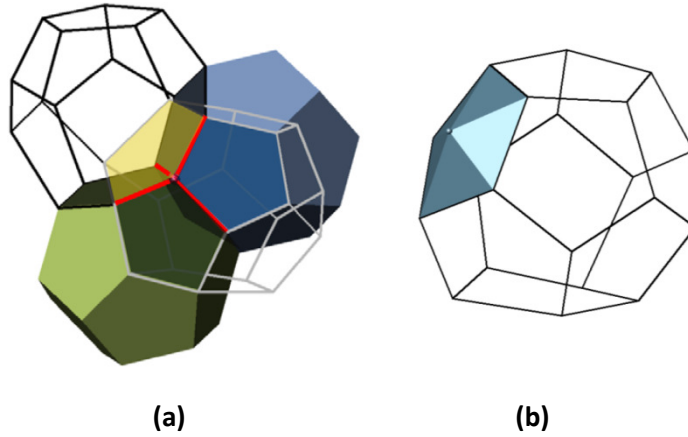


Figure 2.1: FEA cell representation (Yang, 2008)

Forces acting on vertices contribute to deformations and movements of cells. The FEA formulation defines nodes at vertex points, assigning nodal forces based on known physical relationships. Naturally, accurate calculations may only be arrived at if side lengths, face areas and angles are correctly modeled, and this requires that accurate cell-matched 3D reconstructions be available. Figure 2.2 depicts a typical three-dimensional cell arrangement in which various sub-cellular force producing components are shown.

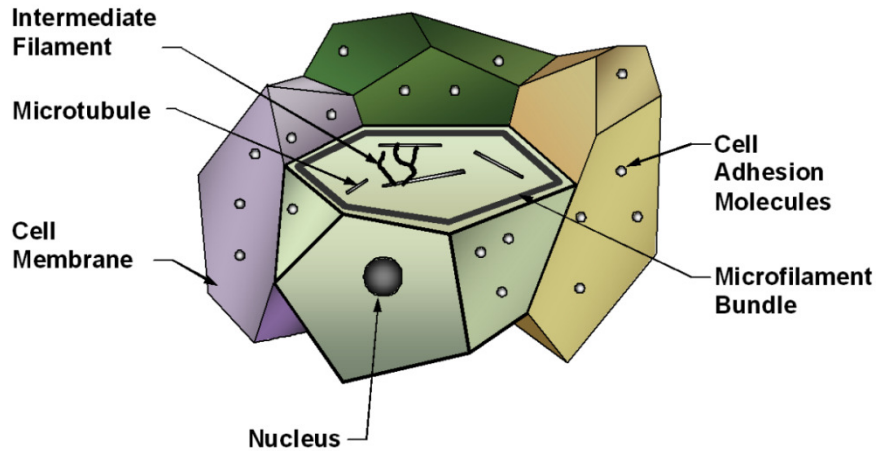


Figure 2.2: Cellular components modelled (Viens and Brodland, 2007)

In the finite element formulation, interfacial tension between adjacent cells is derived from forces generated by these components. This relationship is characterized by Eq. 2.1, where the superscripts '1' and '2' refer to two cells in contact (Brodland, 2002). Note that Eq. 2.1 describes the magnitude of interfacial tension (γ^{12}), as the sum of the magnitudes of various force components.

$$\gamma^{12} = F_{Mem}^1 + F_{Mem}^2 + F_{MF}^1 + F_{MF}^2 - F_{Adh}^{12} + F_{Other}^{12} \quad \text{Eq. 2.1}$$

The term F_{MF} refers to contractile forces produced by microfilaments along the edge of each cell while F_{Mem} describes cell membrane components (including proteins) producing force. Meanwhile,

F_{Adh} corresponds to cell adhesions that release energy when cells come into contact. The term F_{Other} is included to account for other potential force generators (Brodland, 2002).

Physically, Eq. 2.1 describes the interfacial tension that exists on the contacting faces of adjacent cells. Referring to Figure 2.3, interfacial tension is further reduced to nodal forces through Eqs. 2.2 to 2.4, where \mathbf{v}_4 (Eq. 2.3) describes the directionality of nodal forces and Eq. 2.4 describes its magnitude.

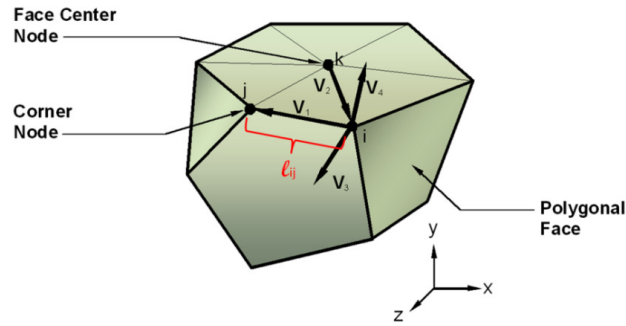


Figure 2.3: Calculation of nodal forces (Viens and Brodland, 2007)

$$\mathbf{v}_3 = \mathbf{v}_1 \times \mathbf{v}_2 \quad \text{Eq. 2.2}$$

$$\mathbf{v}_4 = \mathbf{v}_3 \times \mathbf{v}_1 \quad \text{Eq. 2.3}$$

$$F_i = F_j = \frac{1}{2} \gamma^{12} l_{ij} \quad \text{Eq. 2.4}$$

As seen in Figure 2.3, this is accomplished mathematically by assigning an extra face center node to each face. Constituent triangles are then generated, with vectors along triangle edges utilized to calculate nodal forces. This is repeated for each of the three sides of each triangle, for each face and for each cell. In this way, the total nodal force at each corner node is taken as the sum of 12 force vectors.

Each cell is thus characterized by forces existing on each of its nodes. Furthermore, cells are modelled by 3 sets of orthogonal dashpots similar to that in Figure 2.4. The viscous effects of cell cytoplasm are accounted for by these dashpots (Brodland et al., 2007).

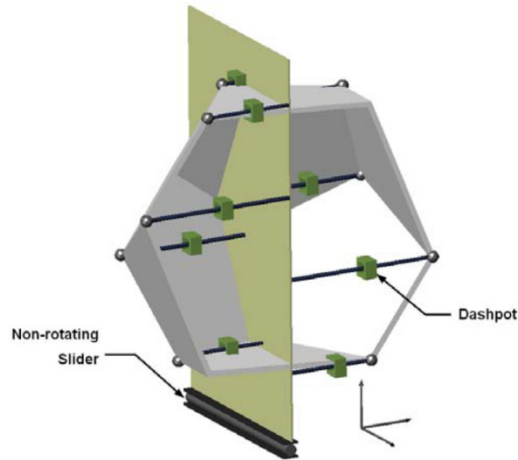


Figure 2.4: Orthogonal dashpot cell-model for single direction (Yang, 2008)

The reader is referred to Brodland et al. (2007) for a complete formulation. Nonetheless, it is the combination of nodal forces and the orthogonal dashpot cell- model that contributes to cell deformations and movements characteristic to dynamic cell-systems.

2.1.2 Model Input Requirements

As this section shows, accurate geometric information aids greatly in obtaining accurate FEA calculations. Without accurate 3D nodal positions, face areas and orientations, the model will not match the physical situation. The first step toward construction of accurate models is the development of a CGA.

2.2 Imaging Modalities

Brief reviews of magnetic resonance imaging (MRI), computed tomography (CT) and fluorescent confocal microscopy (FCM) are provided in this section as they are representative of modern medical imaging modalities. Each modality produces volumetric images, albeit through different processes. The goal is to highlight the difference in information content between images of each type. In conjunction with a discussion on three-dimensional reconstruction methods in Section 2.2.5, it will be seen that existing applications of three-dimensional medical images are insufficient for extraction of the required information for the FEA described in Section 2.1.

2.2.1 Magnetic Resonance Imaging (MRI)

Current applications of MRI technology are plentiful. These range from early detection of breast cancer (Levman et al., 2008) to identification of three-dimensional structures in pre-natal brains (Schierlitz et al., 2001). Other applications include the construction of three-dimensional models of the human tongue, lip and face (Badin et al., 2002) as well as real-time three-dimensional

reconstruction of anatomical structures for pre-operative analysis (Chabererie et al., 1998). Clearly, the potential of MRI in a variety of fields is large.

Figure 2.5 shows a sample MRI image of the human head. Anatomical structures such as the white/gray matter and the skull are clearly discernible. Notice also that different anatomical structures display characteristically different pixel intensities.



Figure 2.5: MRI image of a human skull (“A to Z of tinnitus”)

Methodology (Brown et al., 2004)

Images obtained through MRI result from interactions between atoms in biological tissue and the introduction of; 1) a magnetic field and 2) radiofrequency (RF) energies. In most cases, MRIs target the H^+ atom (i.e. a proton). Fundamentally, any positively charged nuclei (e.g. protons) produce a magnetic field (B_H). When subjected to an external magnetic field (B_o), such positively charged nuclei tend to align with the direction of B_o , either in a parallel or anti-parallel direction. Since directionality and positioning of nuclei are typically disordered and random, the introduction of B_o imposes order and an associated change in system energy.

MRI takes advantage of this by introducing a strong magnetic field on the specimen. H^+ atoms thus reorient to align B_H with B_o . Following this, a series of timed RF energy pulses are introduced into the specimen, with parameters tuned specifically to excite H^+ atoms at localized positions in a volume. RF pulses excite H^+ atoms such that they lose alignment from the magnetic field B_o . Cessation of the RF pulse then causes H^+ atoms to realign with B_o . The realignment of H^+ subsequent to RF pulse cessation is the physical property measured in MRI. As H^+ atoms realign B_H with B_o , a voltage is induced in specially placed transducers. Voltage measured by these transducers is the MRI signal. Since excitations are conducted locally, a complete planar image can be obtained for a particular slice. However, RF pulses are separated by minimal time differences, leading to possible motion artifacts. Correction for these artifacts is not covered here.

MRI images produce high-contrast images. Although H^+ atoms behave uniformly regardless of anatomical position, the particular tissue in which they reside provide “shielding” effects. As a result, H^+ atoms in gray matter are shielded differently from H^+ atoms in the skull. As a result (Figure 2.5), the MRI image intensity in the bone of the skull is much higher than in the gray matter of the brain.

Image Properties

Combination of various RF pulse parameters (i.e. frequency, intensity and timing), results in the localization of MRI readings. MRIs are able to take volumetric readings, and therefore produce three-dimensional images. However, due to hardware tradeoffs and parameters, a full volume is never truly measured. Instead, intermediate values are typically interpolated.

As such, typical MRI outputs are sets of three-dimensional matrices. The three-dimensions correspond with Cartesian x-, y- and z- directions. Careful combination of MRI information results in an accurate depiction of the three-dimensional structure of a specimen. Isolation of particular anatomical structures is conducted utilizing specialized segmentation techniques. Three-dimensional reconstruction methods are discussed in Section 2.3.

2.2.2 Computed Tomography (CT)

CT has been used in a variety of applications. Aside from three-dimensional visualizations, CT has recently been used to aid in cancer research (Paulus et al., 2000), measurement of cortical bone and dental enamel (Spioor Zonneveld and Macho, 1993) as well as medical diagnosis. In many regards, the utilization of CT mirrors that of MRI, as both are three-dimensional imaging modalities.

Figure 2.6 shows a CT cross-sectional image of a human lung. Notice the contrast between various structures in the thoracic cavity. Structures including the vertebral disc are clearly discernible. Unlike MRI images, localization of image properties is not straight-forward. Rather, a complex set of mathematical processes work together to infer information within a CT volume.

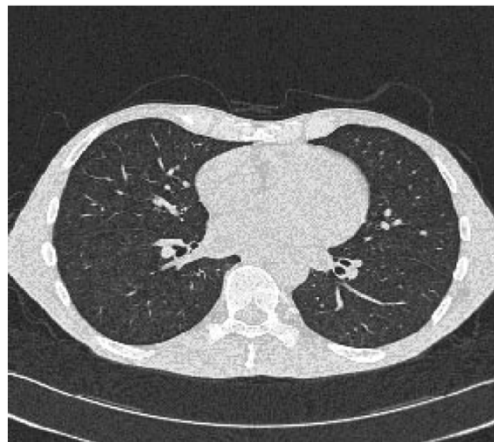


Figure 2.6: CT image of human thoracic cavity (Klingenbeck-Regn et al., 1999)

Methodology

A thorough review of CT methodology is beyond the scope of this thesis. However, the basic approach for generation of three-dimensional CT data will be covered briefly. In 1917, Johann Radon formulated the Radon transform (Eq 2.5) (Ramm and Katsevich, 1996). Fundamentally, it is this transform that led to the rapid proliferation of tomographic techniques in the 1970's and beyond. In

particular, with the advancement of x-ray technologies, CT was made possible through application of the concepts introduced by the Radon transform (Ramm and Katsevich, 1996).

$$Rf(L) = \int_L f(x)d\sigma(x) \quad \text{Eq. 2.5}$$

As related to x-ray projections (and referring to Figure 2.7a), the Radon transform implies that the amount of radiation detected through a specimen is related to the line-integral along the path travelled by source radiation (Iniewski, 2009). That is, the radiation value obtained at the detector can be mathematically manipulated to find attenuation coefficients at any point (x,y) within the specimen. Such manipulation can only be performed if straight-line measurements from many angles and positions in a given plane are taken (Figure 2.7b). The Radon transform thus implies that attenuation of any point of a specimen on this plane can be uniquely determined (Ramm and Katsevich, 1996).

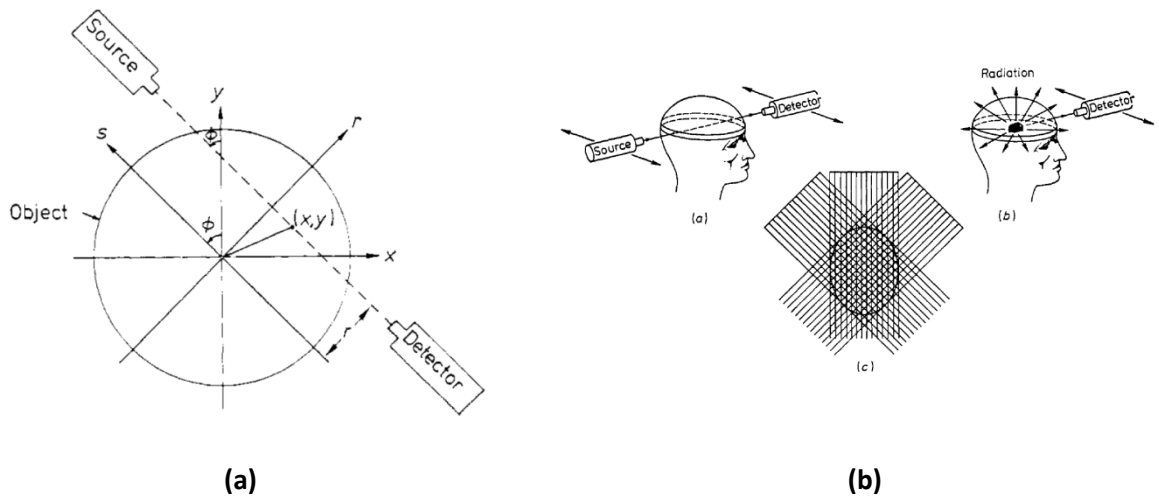


Figure 2.7: CT concept (Ramm and Katsevich, 1996)

This result is profound, as it forms the basis of tomographic three-dimensional reconstructions. CT scanners are therefore designed to obtain x-ray projections of specimens from a wide range of angles and positions. However, practicality dictates that only a limited set of such projections can be obtained. As such, the Radon transform serves only to motivate CT. Rather than explicit utilization of the Radon transform, current CT technologies utilize approximation methods to infer attenuation values within volumes from incomplete data.

Initially methods known as back-projection, filtered back-projection and analytic methods (Brooks and Chiro, 1976) were utilized to compute incomplete attenuation values. However, these were riddled with artifacts, and are no longer commonly utilized. Currently, a popular method for estimating attenuation values within CT specimens is derived from an entropy approach (Gull and Newton, 1986). Although various methodologies exist, the premise is that, given incomplete x-ray

projections, a state of maximum entropy for attenuation values exists. With the application of various constraints and expectations, the maximum entropy state can be chosen from a set of infinite possibilities, leading to the two-dimensional CT image slices seen in modern CT applications. Other variants describe the optimization as that of an error-minimization problem (Wu et al., 2003).

As with MRI, prolonged scanned times render CT images subject to motion artifacts and noise. Furthermore, given the explicit mathematical principles upon which it is built, CT images are inferred from mathematical analysis, and not directly measured. Errors are potentially introduced in this way. Nonetheless, CTs are able to produce high-contrast image volumes similar to MRIs. Tissues with as little as 0.5% differences in density are clearly delineated (Brooks and Chiro, 1976)!

Image Properties

Data obtained from CT scans are also volumetric in nature. However, such information is inferred, as localized readings are calculated through concepts drawn from the Radon transform. Nonetheless, CT data provide strong three-dimensional information upon which visualizations and diagnosis may be produced. As with MRI volumes, CTs may be stored in a three-dimensional matrix. Matrix positions are directly related to physical dimensionality, while values stored at specific matrix positions represent calculated attenuation coefficients. The resulting image thus describes attenuation contributed by biological tissue at a particular physical coordinate. As such, segmentations are performed to isolate anatomical structures based on tissue properties. These may further be used to aid three-dimensional reconstructions (Section 2.3).

2.2.3 Fluorescent Confocal Microscopy (FCM)

Confocal microscopy is an extension to light microscopy. Taking advantage of optics, confocal microscopy obtains near perfect optical slices of translucent specimens (Pawley, 2006). The introduction of specific “fluorescent probes” into the specimen leads to fluorescent confocal microscopy (FCM), where the use of specific wavelengths of light and filters allows for highly targeted imaging (Pawley, 2006). Currently confocal microscopy is utilized in a variety of applications. Typically, these are on the cell-level, with studies on cell-cytology (Kwon et al., 1993) and intra-cellular structures common (Brakenhoff et al., 1985; Lin et al., 2003; Solorzano et al., 1999). Recent applications in ophthalmology and diagnosis of visual ailments have also been developed (Kaufman et al., 2004).

Figure 2.8 depicts a sample confocal image obtained from Antonio Jacinto at the Institute of Molecular Medicine, Lisbon, Portugal, used with permission. The illuminated boundaries represent fluorescent cell membranes from specifically engineered membrane-proteins. Unlike MRI and CT, the entire three-dimensional volume is not imaged. Rather, only those structures exhibiting specific fluorescence are seen.

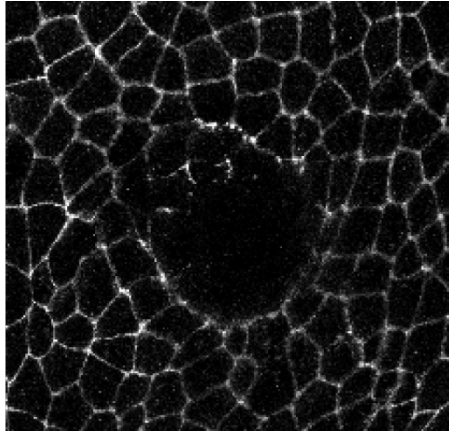


Figure 2.8: FCM image of laser-induced wound in *Drosophila*

Methodology (Pawley, 2006)

Prior to invention of confocal microscopy, the main drawback of light microscopy was the inability to filter unwanted signal from planes other than the focal plane. As such, resulting images invariably appeared blurred due to fluorescence from adjacent planes. With the development of confocal microscopy, this weakness has nearly been eliminated, as confocality allows specific planes within a specimen to be imaged clearly.

In fluorescence microscopy, transgenes are introduced into biological specimens so that specific fluorescent proteins such as green-fluorescent protein (GFP) are synthesized within the cells (Heim and Tsien, 1996). Such proteins react to specific wavelengths of incident light by fluorescing at a different energy level. The fluorescence of these proteins constitutes the signal of fluorescence microscopy, as it may be recorded by imaging devices. However, since fluorescence occurs for all proteins within the specimen, the image is blurred by out of plane fluorescence. FCM alleviates this problem by applying the principle depicted in Figure 2.9. By focusing the fluorescent signal with a pinhole, the resulting image is largely filtered of out of plane fluorescence (Pawley, 2006), resulting in a clear optical slice image.

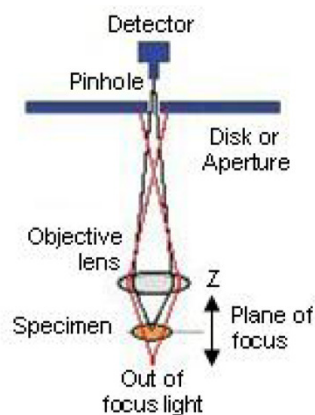


Figure 2.9: Confocal optics (Groh, 2008)

Contemporary confocal microscope systems utilize a laser as the light source (Groh, 2008). This is advantageous, as it allows for targeting of specific wavelength reactive proteins. Furthermore, laser light is highly concentrated, eliminating artifacts due to diffraction. Indeed, there exist systems utilizing more than one laser, allowing for simultaneous imaging of differently labeled cellular structures. The power of confocal microscopy is clear, as it allows live biological specimens to be examined on the cell-level.

Image Properties

Image data obtained from confocal microscopes are directly related to the intensity of fluorescence. Due to the ability to target particular optical planes, image sets may be volumetric, accounting for the entire volume of a particular specimen. However, note that because fluorescence is due to the presence of genetically modified proteins, it is possible that certain structures or cells within the specimen are not imaged clearly. That is, biological variability potentially plays a role in the quality of confocal images.

Once again, confocal image data may be represented in a three-dimensional matrix, with intensity values being stored at each location. It is important to note that the intensity values correspond to the fluorescence of protein at the particular position, and may not correlate to the presence of cellular structure. Intensity values must therefore be interpreted with care, as imaging artifacts may be difficult to explain.

2.2.4 Comparison of MRI, CT and FCM

The discussion provided in Sections 2.2.1 through 2.2.3 shows clear differences between MRI, CT and FCM. These are summarized in Table 2.1.

Table 2.1: Comparison of MRI, CT and FCM

	MRI	CT	FCM
Image Level	Anatomical	Anatomical	Cell
Principle	Magnetic resonance	Radon transform	Light microscopy
Relationship	Physical	Mathematical	Physical
Signal	Tissue shielding	Tissue attenuation	Protein fluorescence

MRI and CT are both medical imaging modalities in which properties of biological tissue allow imaging of gross anatomy. Note that MRIs physically measure the shielding of tissue at localized positions while CTs infer tissue attenuations through a set of mathematical procedures. FCMs on the other hand measure fluorescence of genetically modified fluorescent proteins, and are subject to biological variability. Additionally, it is seen that MRI and CT are highly specialized in delineating anatomical structures within biological specimens. However, FCM operates on a smaller scale, and provides images of cellular structures much smaller than the tissues imaged by MRI and CT.

From this perspective, FCM has little in common with MRI and CT. MRI and CT are used mainly as diagnostic tools with regards to anatomy, whereas FCM largely remains a specialized tool

for research purposes. Additionally, MRI and CT function on the anatomical level, while FCM functions on the cell-level. However, given that all three modalities produce data sets stored in three-dimensional matrices, one might expect that visualization and reconstruction methods utilized on MRI and CT sets might be applicable to FCM. This turns out to not be the case, for reasons explained in the balance of this chapter.

2.2.5 Three-Dimensional Reconstructions

In general, there are three methods for visualization of volumetric data. These are: 1) slice-views, 2) isosurface extraction and 3) volume-rendering (Girod et al., 2000). Slice-views are achieved by sectioning planes within a volumetric data set, with output being the display of two-dimensional sections. Isosurface extractions are more commonly known as segmentations, and are particularly useful in medical imaging since tissue properties imaged by MRI and CT are largely separable. Volume rendering refers to a mathematical inverse of the CT process, whereby projections from arbitrary distance and angle are inferred by the information present in a three-dimensional volume (Girod et al., 2000).

In truth, only segmentations generate true three-dimensional information. That is, slice-views provide accurate two-dimensional representations of specific planes within a volume, but lack three-dimensional context. Additionally, the process of volume rendering is essentially the reverse of CT image formation, creating images with inferred depth through textures and colours. On the other hand, segmentations inherently acknowledge three-dimensional connectivity, allowing for true three-dimensionality.

As such, this section outlines the main principles underlying image segmentations with respect to MRI and CT. The reconstruction of segmented images is obtained through surface reconstruction algorithms, and will also be discussed. The purpose of these discussions is to present the state-of-art for three-dimensional reconstructions of volumetric images. Furthermore, why these techniques are not suitable for cell data will be explained.

Segmentation

As described previously, MRI and CT measure direct tissue properties. MRI contrast is contributed by tissue shielding, while CT contrast is a result of tissue attenuation. In this way, biological structures of differing tissues are clearly contrasted in the images obtained. Furthermore, given that contiguous anatomical structures are typically made up of similar tissues, specific structures may be delineated from others by means of segmentation.

Segmentations may be constructed in a variety of ways. The simplest is with iso-surface extraction. This method utilizes simple thresholds to define the range of values that are to be accepted as part of specific structures (Girod et al., 2000). Other techniques utilize image information (e.g. gradients) to infer connectivities (Cline et al., 1987) from suitable starting points while atlases defining expected shapes may also be used as references (Fisher et al., 1997; Collins et al., 1995). Additionally, it has been demonstrated that features extracted from images may be used

in producing secondary measures for segmentation (Jones and Metaxas, 1997; Sarti et al., 2000; Udupa, 1982). Intense study is currently underway in this area, as it potentially leads to automated diagnosis and location of tissue aberrations from MRI or CT data (Levman et al., 2008; Magoulas and Prentza, 2001)).

Reconstruction

Given a segmented anatomical structure from MRI/CT, it is possible to create a surface reconstruction (Figure 2.10) from known sample points (Amenta et al., 2000; Bernardini et al., 1999; Boissonnat and Cazals, 2000).

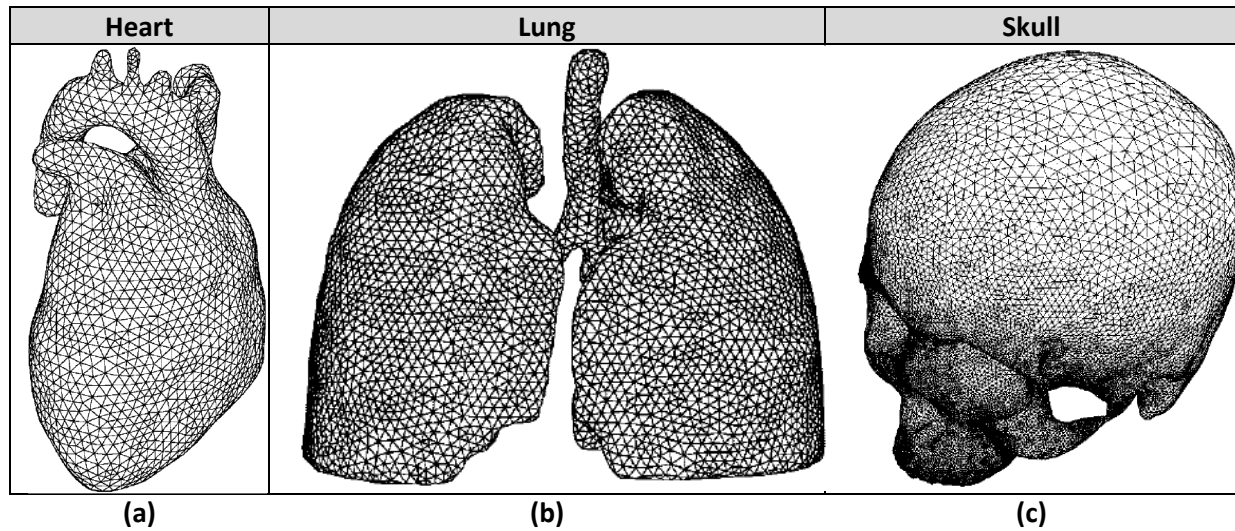


Figure 2.10: Sample Delaunay surface reconstructions (Wang et al., 2006)

The segmented structure inherently contains coordinate information regarding the location of the anatomical structure in question. As such, methods such as marching cube algorithms (Nielson, 2003), Delaunay triangulations (Amenta et al., 2000; Boissonnat and Cazals, 2000; Amenta, Bern and Kamvyselis, 1998) and ball-and-point algorithms (Bernardini et al., 1999) may be used to produce surface representations of the structure. These surfaces form the outer layer of the segmented structure, and provided accurate three-dimensional representations.

Such techniques have been used extensively in the medical community. Applications range from reconstruction of blood vessels (Barillot et al., 1985) to reconstruction of cortical bone (Stout et al., 1999). Figure 2.10 shows samples reconstructions of a human heart, lung and skull respectively (Wang et al., 2006). Although powerful, these methods are not suitable for extraction of a three-dimensional cell-level FEA. Meshes generated are fit arbitrarily to surface characteristics of the object in question, and provide no information on the smaller components of each object. For example, in Figure 2.10a, the aorta is not separable from the rest of the heart. Referring this to reconstructions of biological cells, existing reconstructions do not allow the separation of individual cells within a mass.

Furthermore, since FCM does not measure specific tissue properties, segmentation accuracy in confocal data does not compare to that of MRI or CT. There exists an abundance of literature dealing with confocal segmentations (Solorzano et al., 1999; Dima et al., 2002), but these are largely specialized, and depend upon the particular cellular structure of interest. Examples include the segmentation of cell nuclei (Solorzano et al., 1999), the antennal lobe of *Drosophila melanogaster* (Laissue et al., 1999; Lin et al., 2007) and neurons in the brain (Dima et al., 2002; Al-Kofahi et al., 2003).

Depending on the research interests of particular researchers, FCM methodology allows many cellular components to be targeted specifically through genetic modification. The most common of such structures are the cell nuclei and cell membrane. Fundamentally, techniques applied to cell nuclei segmentation do not apply to cell membrane extraction. Within the context of the FEA described in Section 2.1, a mere extraction of cell membrane location is insufficient in initializing a 3D FEA analysis. That is, individual cell adjacencies cannot be established by knowing the location of cell membranes. Each cell must be individually labelled and analyzed for adjacency information.

As such, although it is conceivable that surface reconstructions can be obtained for extracted cell membranes, these lack information required by the 3D FEA. Additionally, since protein fluorescence is subject to biological variation, signal intensities seen in confocal images may not be as readily segmented as tissue properties in MRI and CT. On a slice-by-slice basis, photo-bleaching (Pawley, 2006) may occur in FCM, causing signal intensity discontinuities, and potential failure of the 3D segmentation process. With these facts in mind, it is clear that existing methods in medical imaging are not suitable for reconstruction of cell-level data.

Here, closed contours within each slice are considered individual cell outlines, and will be associated with contours in adjacent slices, if appropriate. The goal is to obtain a cell-by-cell listing of the contours that define it.

2.3 Data Clustering and Optimization

The methods utilized to obtain contour associations mirror closely the machine learning methods utilized in medical imaging. That is, from FCM image stacks, association of contours requires the extraction of features from images. Additionally, features will be processed utilizing clustering and optimization techniques to obtain contour matches. Thus, the approach presented here utilizes methodologies similar to that used in standard medical image processing techniques.

Measurements obtained from images may be used to guide data clustering or machine learning algorithms. Here, data clustering refers to the process of computationally dividing data into accepted classes. For example, a study by Levman et al. (2008) suggests that DCE-MRI images of breast lesions may be analyzed using support vector machines to classify malignant and benign tumors. Analysis is based upon mathematical properties derived from MRI images, and is

representative of typical data clustering applications. Most applications seek underlying patterns or similarity within source data.

Currently, this field of research is extremely active, with the development of expert systems and automated diagnoses being of particular interest (Magoulas and Prentza, 2001). Applications range from interpretation of ECGs (Magoulas and Prentza, 2001) to early detection of breast-cancer (Levman et al., 2008). Development of such algorithms typically requires the assistance of an expert (e.g. medical practitioner), providing rules and knowledge through which computational methods may interpret data. However, it is often found that experts may not truly understand the basis of diagnoses. That is, exact rules governing differences between malignant or benign lesions in imaging may not be available (Magoulas and Prentza, 2001). As such, decisions in expert systems are made with respect to image properties through evaluation of an objective function. The challenge is therefore to obtain a natural set of information that may aid in decision making.

In the context of this thesis, data obtained from FCM must be carefully analyzed to obtain suitable information to guide a contour grouping algorithm (CGA). As with algorithms developed for medical diagnoses, the contour grouping algorithm is highly specialized. The goal is to arrive at a set of correctly associated contours within a FCM stack of cells. Due to the biological nature of FCM images, useful information may not be apparent at first glance. There is a need to extract a set of image features, direct or derived, effective in guiding contour groupings. The reader is referred to Magoulas and Prentza (2001), Jain et al.,(1999) and Milligan and Cooper (1987) for a full review on existing clustering methodologies.

In general, the task of feature extraction (Figure 2.11) is to obtain a subset of m features from a total of d possible features (i.e. $m \leq d$). Each data point (i.e. pixel) is represented by a feature vector, containing the values of all m features. As mentioned, the feature extraction is characterized by the minimization of a derived objective function (Jain et al., 1999). From a mathematical standpoint, this process reduces the search space for objective minimization from d -dimensions to m -dimensions. This serves two purposes, as feature-space reduction minimizes problem complexity, while choice of suitable features allows algorithms to only consider relevant variables.

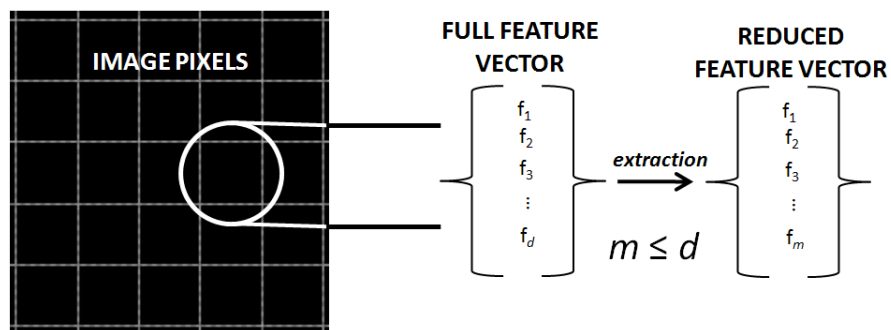


Figure 2.11: Feature Extraction

In many clustering applications, Euclidean distance or normalized distance classifiers are used. The motivation lies with the idea of dimensionality, as distance features lend themselves to interpretation in Euclidean n -space. As such, classical methods of data clustering look to minimize the “distance” between and within clusters (i.e. inter- and intra- cluster). Examples include K-means clustering and nearest neighbour (NN) clustering (Jain et al., 1999). For these methods, distance is defined as a function of selected features, typically as a variant of mean-squared error along normalized feature axes.

Other clustering methods include fuzzy clustering, neural network clustering and genetic clustering (Jain et al., 1999). Fundamentally, these methods are only slightly different than K-means and NN, but arise from diverse concepts. Fuzzy clustering may be applied to any clustering methodology, but includes the consideration that each data point may have partial membership to any number of clusters. Neural network clustering bases decision making on mathematical models describing human learning (e.g. utilization of learning data). Features are input into a complex system of pre-defined mathematical operators, whereby the final system output determines cluster membership. Genetic clustering models decision making to the random assortment of genes, and is an iterative algorithm whereby solutions are evaluated via a fitness function (defined by the user). Successive, seemingly random, solutions are proposed and evaluated. If the solution is insufficient (with respect to fitness), the process is iterated with a new, re-assorted solution.

Note that neural network and genetic algorithms have also been proposed in solving the “Traveling Salesman Problem” (TSP) (Peterson, 1990). Thorough reviews of these and the TSP can be found in Peterson (1990) and Larranaga et al. (1999). Other methods proposed in solving the TSP include simulated annealing (Malek et al., 1989), and an elastic net algorithm (Durbin et al., 1989). In this regards, it is seen that data clustering falls within the larger field of combinatorics. As such, the CGA need not confine itself to mirroring decision methodologies applied to medical imaging. Rather, the field of combinatorics may also provide suitable motivations in developing the CGA.

Particularly, graph partitioning is often applied to problems of network optimization and parallel computing (Hendrickson and Kolda, 2000). Such applications are similar to contour grouping since the goal is to minimize an objective function by partitioning data points into n partitions (Zha et al., 2001). Additionally, graph partitioning utilizes nodes to represent data points, with nodal association costs driving optimizations. That is, each node in the data set may be associated with any other node. However, each association contributes a different “cost of association”, and therefore is the main factor in driving overall partitioning cost. Thus, local association costs are seen to drive global optimization of partitioning.

The chief difference between current graph partitioning applications and the CGA is that network optimizations or parallel processing cases define n partitions required after optimization. In the case of contour groupings, the n number of cells in the FCM stack is assumed to be unknown. Though subtle, this difference renders graph partitioning methods very difficult to adapt to the contour grouping problem. However, local association costs are seen to be beneficial, as it allows for the definition of specific features for association of contours.

From the above discussion, it is clear that a large variety of clustering algorithms and optimization methods exist. These have been applied extensively in many fields and are currently heavily applied in medical imaging applications. Identification of cancerous tissue, isolation of anatomical structures and real-time image reconstructions are but a small sampling of such applications. With respect to this thesis, clustering and optimization methods will be used to motivate the grouping of contours from one FCM slice to the other.

However, the CGA does not strictly adhere to existing methodology. Rather, the ideas of feature extraction, clustering and optimizations have been synthesized in creating a novel grouping framework. Particularly, the feature-space has been reduced from a global-scale to a local-scale. That is, features (from here on referred to as metrics) for a particular contour need not describe its relationship with those in all slices. Metrics need only define the relationship between a contour and those in adjacent slices. This greatly reduces complexity of the problem, while allowing for the definition of specific association metrics similar to those used in graph partitioning. In a strict sense, clustering has been reduced to a micro-level decision making process between contours in adjacent slices, with macro-level outputs being the grouping of contours into clusters representing true three-dimensional cells.

Chapter 3: Contour Grouping Algorithm

With the grouping and clustering techniques from Chapter 2 in mind, the discussion shifts towards the contour grouping algorithm (CGA) developed for this thesis. First, a suitable set of metrics is defined. These are explicitly defined in the following sections. Metrics are then incorporated into a composite contour grouping algorithm that has proven, through various trials, to be accurate. This composite approach synthesizes ideas from probability theory and optimization methods. Composition of these ideas leads to a two-part grouping algorithm.

The CGA is meant to bridge the gap between confocal image stacks and three-dimensional finite element analyses. As demonstrated by Brodland and Chen (2000), finite element analysis is a powerful tool that, in conjunction with real- images, may be used to obtain a complete physical description of biological interactions.

3.1 Source Data

Development of the CGA required utilization of a synthetic cell aggregate (Figure 3.1) created with custom software from the Biomechanics Lab at the University of Waterloo. Cells isolated from the aggregate (Figure 3.2) were analyzed to obtain information on metrics for contour association.

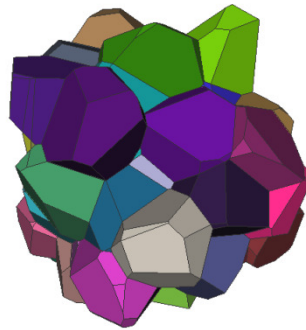
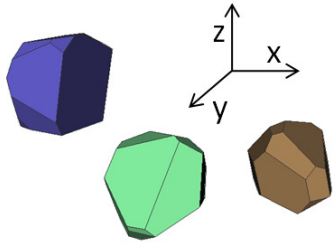


Figure 3.1: Cell aggregate

Isolated Cells (viewpoint 1)



Isolated Cells (viewpoint 2)

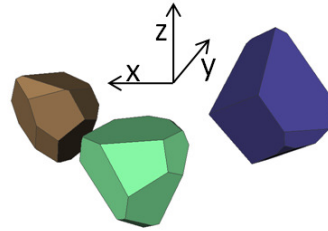
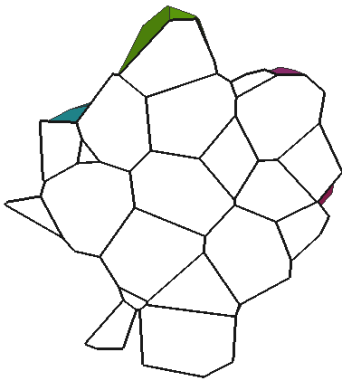


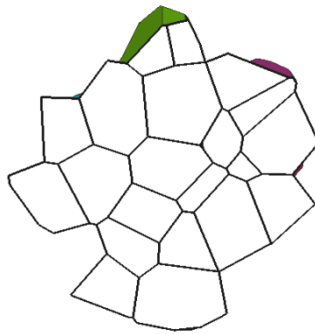
Figure 3.2: Isolated cells 3D view

Note that Figure 3.2 depicts the three-dimensional shape of isolated cells and not their confocal representations. To simulate confocal image stacks, the aggregate was artificially cut at evenly spaced intervals along a particular axis. Image snapshots taken after each cut resulted in images similar to Figure 3.3. Cells lying on the plane-of-cut are seen as white areas enclosed by a solid black contour. These contours represent the “in-slice regions” that are of interest to the CGA. Coloured cell faces have not been removed to provide a sense of depth. In truth, confocal slices only show in-slice contours.

Sample slice 1



Sample slice 2



Sample slice 3

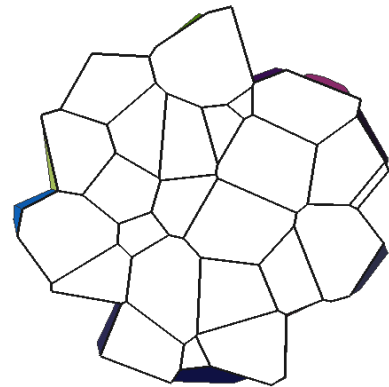


Figure 3.3: Sample slices

Figure 3.4 depicts a single isolated cell (represented by contours) from the aggregate in Figure 3.1. Contours have been obtained from simulated confocal slices and are manually associated across slices. Notice that information between slices is lost. Centroids of each contour are also plotted in red, along with vectors connecting centroids of adjacent slices in green. In examining Figures 3.4 (a) and (b) respectively, it is clear that two-dimensional image slices only give reasonable three-dimensional cell representations if contours are properly associated across slices.

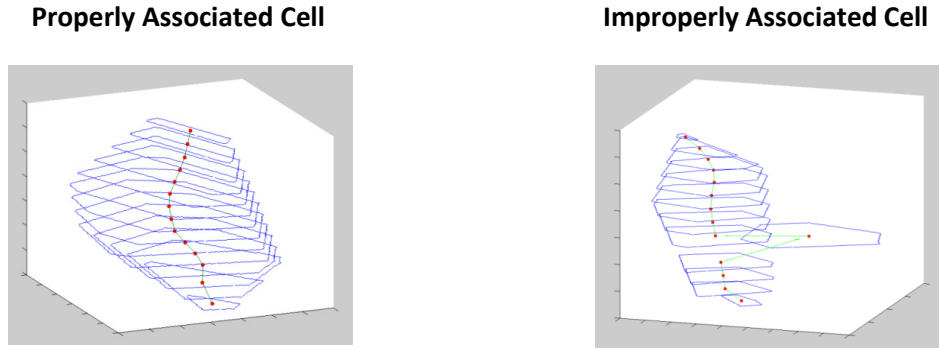


Figure 3.4: 3D region associations

Thus, the issue at hand is to determine which contours in different slices may be associated to form accurate cell geometries. The chief difficulty lies with identification of metrics determining contour similarity. These metrics must be derived from confocal image slices, relying on information extracted from confocal image sets. The remainder of this chapter discusses specific image properties and the definition of various metrics.

3.2 Image Properties and Metrics

The CGA requires accurate extraction of particular image properties. Without discussing specific implementations, confocal image stacks are analyzed with image processing techniques to obtain; 1) slice contours and 2) contour centroids (Table 3.1).

Table 3.1: Image information

Extracted Information
Contour boundary
Contour centroids

That is, for each slice, closed boundaries are traced to obtain a contour. The centroid of each contour is subsequently calculated. From this information, a number of derived quantities can be obtained (Table 3.2). These quantities fall within the categories of: 1) geometric properties and 2) coordinate properties.

Table 3.2: Derived and measured properties

Property	Derived From
<i>Geometric Properties</i>	
2D contour shape	<ul style="list-style-type: none"> • Contour boundary • Contour centroid
Contour area	<ul style="list-style-type: none"> • Contour boundary
<i>Coordinate Properties</i>	
Contour position	<ul style="list-style-type: none"> • Contour centroid
Contour neighbours (adjacent slices)	<ul style="list-style-type: none"> • Contour centroids
Contour offset	<ul style="list-style-type: none"> • Change in centroid position across slices of associated contours
Contour topology (same slice neighbourhood)	<ul style="list-style-type: none"> • Contour boundaries • Contour centroids

Assuming image qualities and processing techniques are ideal (as is the case with synthetic data), the goal is to identify quantities to be used in guiding contour associations across slices. Furthermore, due to the three-dimensional nature of confocal image stacks, both inter-slice and intra-slice metrics exist. Thus, careful analysis of individual contours, their derived properties and relationships with inter- or intra- slice neighbours will lead to identification of valid association metrics. For the remainder of this section, properties will be analyzed in context with previously manually associated contours. Additionally, suitable properties will be recast into suitable metric definitions.

3.2.1 Geometric Properties: 2D Region Shape

The first geometric property to be examined is region shape. The shape of a traced contour may be modeled by an equivalent ellipse with aspect ratio κ . Aspect ratio is calculated from Eq. 3.1, utilizing moments of area.

$$\kappa = \sqrt{\frac{I_{yy}}{I_{xx}}} \quad \text{Eq. 3.1}$$

In previous studies (Chen and Brodland, 2000; Brodland and Veldhuis, 2002; Brodland and Wiebe, 2004; Yang and Brodland, 2009), the aspect ratio has been described as an important cell property. Specifically, κ is related to specific cell mechanics. Generalization of contours into equivalent ellipses therefore avoids difficulties in describing arbitrarily shaped contours, while maintaining physical correspondence to mechanical properties. In terms of the CGA, is it possible to utilize equivalent ellipses to guide contour associations?

Equivalent ellipses may be constructed from individual contours and their centroids using calculation of moments of area. Once obtained, it is trivial to find an aspect ratio (κ) for each ellipse

(Eq. 3.1). Each manually associated contour group will thus possess a list of aspect ratios. Figure 3.5 shows the three-dimensional slice profile of a contour group and its equivalent ellipses. Notice that centroid positions are identical since equivalent ellipses are defined with respect to centroid location. However, by inspection, overall three-dimensional shape is distorted due to approximations made in arriving at equivalent ellipses for each contour.

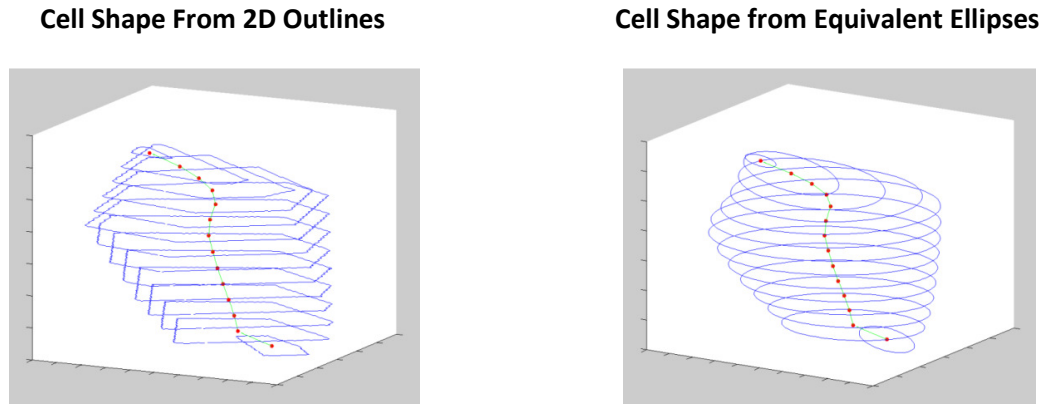


Figure 3.5: Cell shape and equivalent ellipses

Analysis of aspect ratios proceeds through evaluating Gaussian distributions of κ for manually associated contour groups. That is, considering the κ -values of individual contours in each group, a mean (μ) and standard deviation (σ) can be calculated. Taking these as parameters in a Gaussian distribution and normalizing, it is possible to compare the shape profiles of each contour group. This is shown in Figure 3.6. As evidenced by the clustering of distributions, each contour group has similar κ -value distributions. For each group, the mean value is quite similar, while the variance differs. Given arbitrary κ -values, it is thus difficult to establish contour associations since distinct groupings do not exhibit characteristic Gaussian shapes. It seems unlikely that κ is a feasible metric to be used in the CGA.

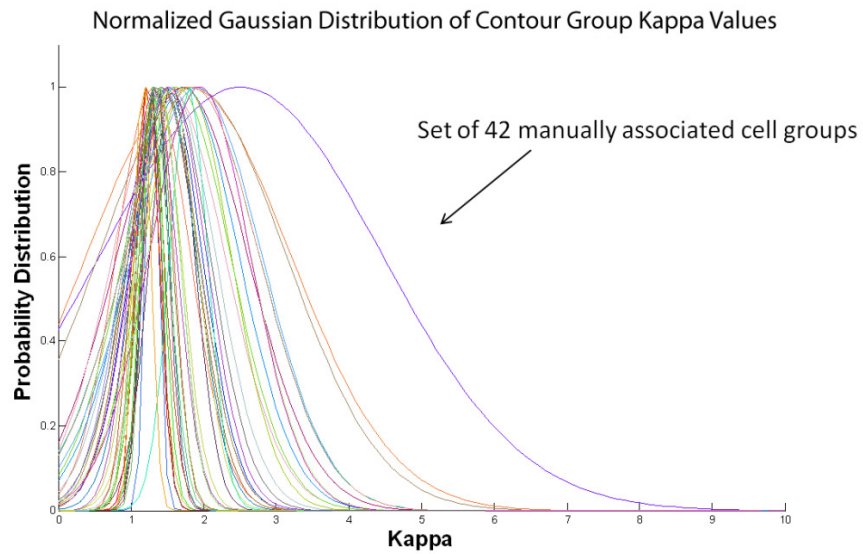


Figure 3.6: Distribution of cell kappa values

Furthermore, Figure 3.7 shows the distribution of κ in terms of equivalent ellipses within individual slices. Once again, separation is not possible, indicating that contours within a slice do not vary significantly in shape within the image stack. Taking these observations into account, it is stated that aspect ratio (κ) of an equivalent ellipse does not provide information useful for the CGA. Since it is not possible to distinguish between distributions of κ for individual groups and that of regions within specific two-dimensional slices, development of a metric is fruitless. The two-dimensional shape of contours do not provide useful information regarding possible contour associations.

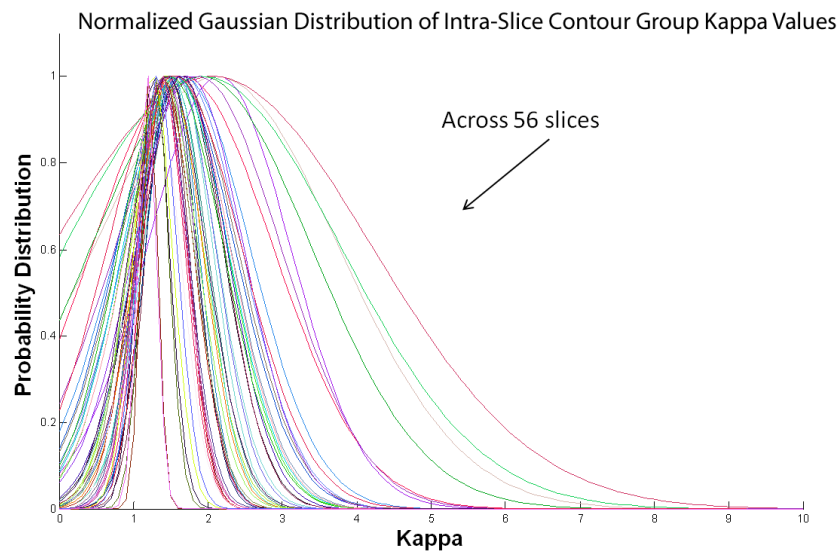


Figure 3.7: Distribution of cell kappa values within slice

3.2.2 Geometric Properties: Region Area

The second geometric property examined is region area. With knowledge of contour boundaries, the amount of “area overlap” between regions on adjacent slices can be found. Consider the isolated three-dimensional cell shown in Figure 3.8(a). Figure 3.8 (b) through (h) depict slices obtained at regular intervals. The white regions in each image indicate the contour seen on the cutting-plane, parallel to the plane of text.

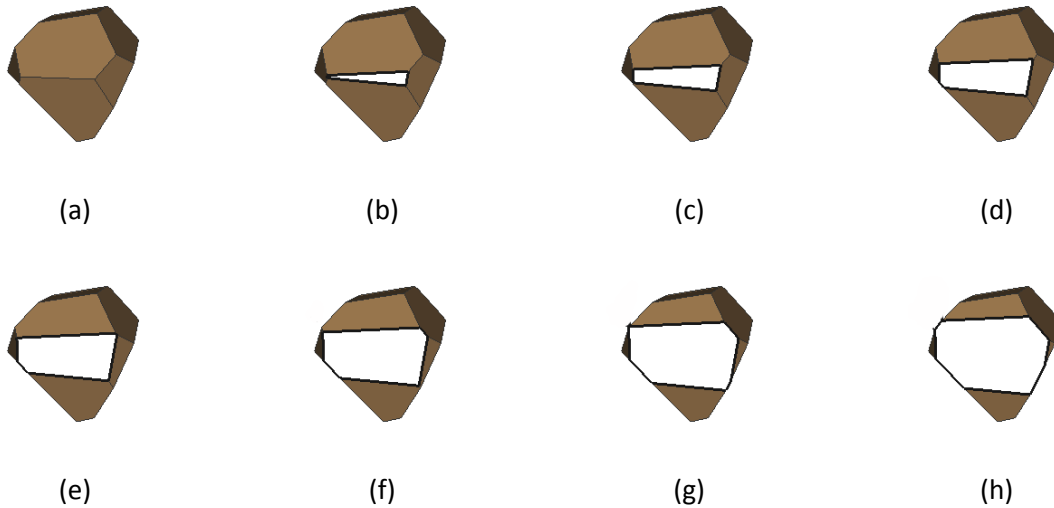


Figure 3.8: Slice 3D cell

Superimposing cutting planes of Figures 3.8(b) through Figures 3.8(h), the image in Figure 3.9 is obtained. As the cutting plane progresses depth-wise in the direction normal to the page, successive contour areas show a large measure of overlap. Therefore, it is seen that properly associated contours exhibit a significant area of overlap between adjacent slices.

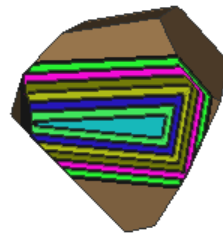


Figure 3.9: Superimposed region areas

Metric Definition: Pixel Overlap

In digitized images, area overlap may be defined by the number of overlapping pixels between contours of adjacent slices. As seen in Figure 3.10(a), through projection of contour areas (A1 and A2) from adjacent slices onto the same plane, an overlap area (white) is seen as intersecting pixels between A1 and A2. The number of overlap pixels are counted and stored as the pixel overlap of the

contours in question. Note that this number does not provide a clear measure of similarity between two contours. Rather, it is necessary to normalize the number of overlapping pixels with respect to the initial size of compared contours. This results in a measure of overlap that may be compared between various regions.

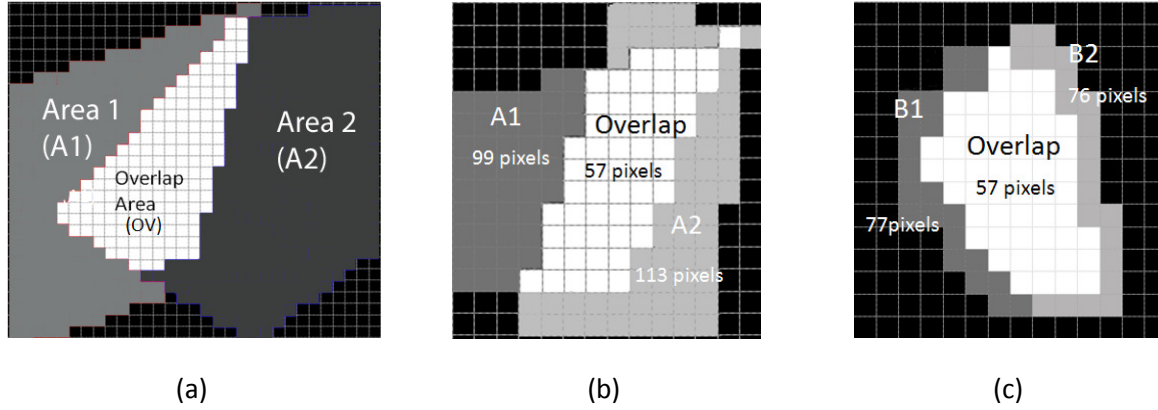


Figure 3.1: Pixel overlap

Figure 3.10 (b and c) illustrates the need for overlap normalization. In (b), the comparison regions are larger than those in (c). Despite an identical number of overlapping pixels, it is erroneous to state similarity of regions A1 and A2 in Figure 3.10 (b) is equal to that of regions B1 and B2 in Figure 3.10 (c). Normalization accounts for this difference. The normalization process utilizes percentage overlap of each contour, described in Eq. 3.2 and Eq. 3.3. These percentages are treated as scaling factors in Eq. 3.4, the final normalization equation.

$$\%_{overlap A1} = \frac{OV}{A1} \quad \text{Eq. 3.2}$$

$$\%_{overlap A2} = \frac{OV}{A2} \quad \text{Eq. 3.3}$$

$$overlap\ metric = \frac{A1}{A1 + A2} (\%_{overlap A2}) + \frac{A2}{A1 + A2} (\%_{overlap A1}) \quad \text{Eq. 3.4}$$

Briefly, normalization assigns strength to individual overlap percentages based upon relative amount of pixel area each contour encloses as compared to the sum of both contour areas. In doing so, normalization scales the overlap percentage of regions A1 and A2 to a value in the range of [0, 0.5]. By summing these scaled values (Eq. 3.4), the overall metric thus lies on the interval [0, 1], with equal weighting given to each contour, regardless of size. In this way, the pixel overlap metric can be interpreted as a measure of percentage similarity.

As an example, consider the two cases presented in Figure 3.10 (b) and (c). Table 3.3 summarizes metric values obtained for both cases. Based upon these values, contours B1 and B2 are more similar than A1 and A2. Visual inspection confirms this result, suggesting that the metric is successful in both normalizing regions of different size and in providing a meaningful measure of similarity. As a final note, it is emphasized that the pixel overlap metric is defined over two regions (and slices). Thus, it is a “pair-wise metric”.

Table 3.3: Example pixel overlap metric

	Area 1	Area 2	OV	Overlap Metric
a)	99	113	57	0.5424
b)	77	76	57	0.7452

3.2.3 Coordinate Properties: Region Position and Neighbours

The position of a contour is defined as the position of its centroid with respect to the image stack. The relative position of each contour is defined by a three-dimensional vector $P = (x, y, z)$. In general, the x- and y- coordinates are integers corresponding to the row and column indices of a given image slice, while the z- coordinate denotes the slice upon which it lies. From an imaging standpoint, non-integer values do not exist, since images are represented as discrete pixels. However, for this thesis, sub-pixel values are allowed since the vector P may be interpreted as lying in Cartesian coordinate space with continuous coordinate values.

Definition of centroid position with respect to Cartesian space allows for considerations of contour neighbourhoods. That is, for each contour, a list of neighbors can be built based on centroid distances. When considering contour positions, the obvious assumption is that correct associations favour contours close to one another. Figure 3.11 depicts a particular contour and its four nearest neighbours in each adjacent slice (i.e.: previous and next slices). With vectors drawn from the contour centroid to each neighbor, the reader is asked to imagine rotation of this figure about the z-axis. If one considers the magnitude of these vectors in three-dimensional Cartesian space, it becomes apparent that distance alone is not sufficient in defining a metric for associations (as each vector is quite similar in magnitude). Indeed, if one were to apply a classical K-means classifier (assuming the number of cells ‘K’ was known) with distances as metric, correct associations are not made. Certainly, the K-Means algorithm may be customized with other information to guide it towards correct associations, but this approach is not examined here.

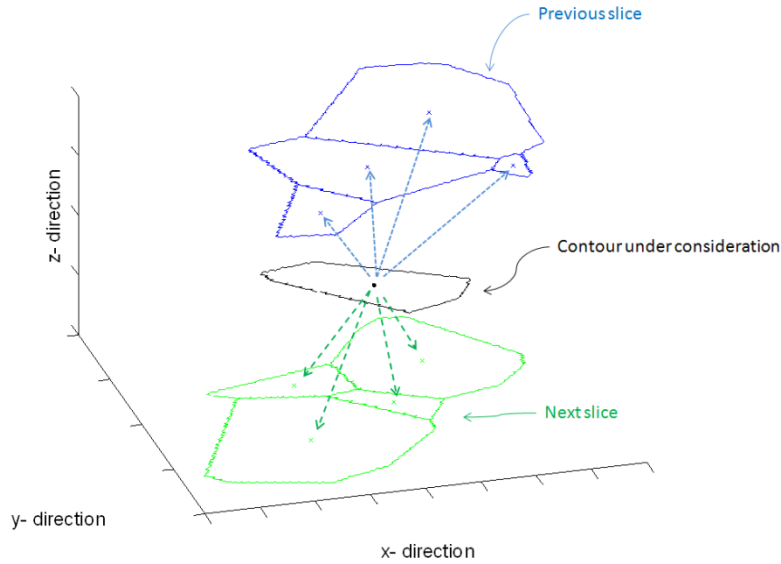


Figure 3.2: 4-nearest adjacent slice neighbours diagram

Rather, the current analysis has provided an important observation for computational purposes. Suppose a list enumerating the n -nearest neighbours of a contour in adjacent slices is created. Each contour will be, at most, assigned an adjacent slice neighbourhood with $2n$ neighbours. Assuming that correct contour associations lie within this list, then computational complexity of the CGA is reduced dramatically. Consider the diagram depicted in Figure 3.11. A list of 4-nearest adjacent slice neighbours results in a list of 8 neighbours. In comparison to the case depicted in Figure 3.12, in which all adjacent slice neighbours (35 in total) are enumerated, the potential for computational reduction is seen. For this particular case, assume that the current slice has 18 distinct contours. A neighbourhood of 8 contours results in a full list of 144 entries. Compared with a total of 630 entries when all adjacent slice contours are enumerated, this represents a large savings in both computational and memory requirements, especially since lists are needed for each contour within all slices (e.g. $[144 \times 21 \text{ slices}]$ vs. $[630 \times 21 \text{ slices}]$).

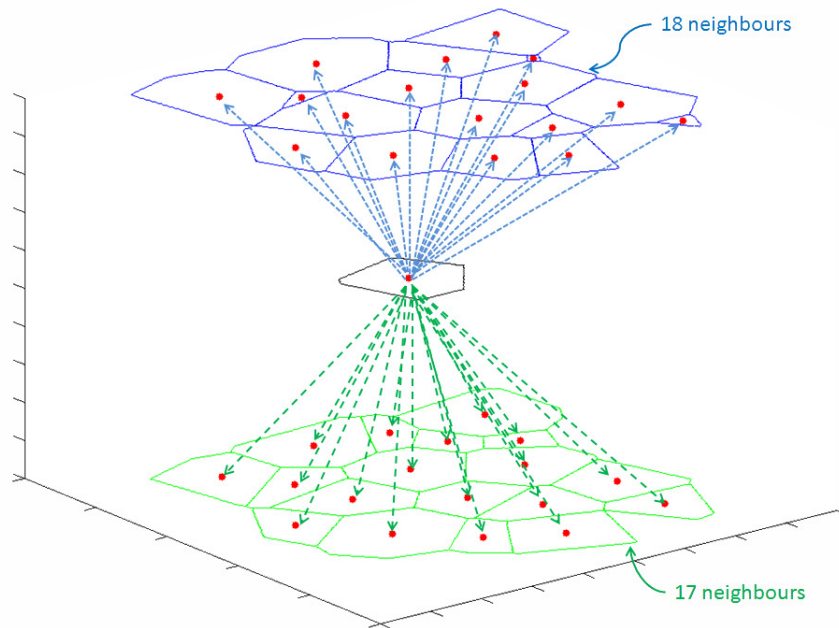


Figure 3.3: Diagram of all adjacent slice neighbours

In this way, contour position may be used to reduce the complexity of the CGA. From this point forwards, the position of a contour will be defined relative to its eight-nearest neighbours on adjacent slices. The assumption is made that correct associations across slices exist within this list (if at all).

3.2.4 Coordinate Properties: Region Offset

Figure 3.4 indicates that correctly associated contours are not uniform in shape. Since changes in contour shape drive offsets in contour position (i.e. centroid position shift), \mathbf{P} may be used to explore displacement. That is, from slice-to-slice, the difference between \mathbf{P} vectors may be used to define centroid displacement on the x-y plane. Unfortunately, if we are given contours in three-dimensional space without predefined associations, it is impossible to truly define displacement (since we do not know which vectors are to be subtracted). Thus, the following discussion is based upon a set of manually associated regions across three adjacent slices (Figure 3.13). Additionally, it is not possible to draw information from individual displacement vectors, as each vector may take on a large number of orientations and magnitudes. Potential information drawn from displacement thus lies with definition of an overall vector field. If directionality of all displacement vectors between adjacent slices is similarly oriented, it is implied that centroid displacement may be following a specific pattern. Contour associations can thus be guided by “looking” in this particular direction.

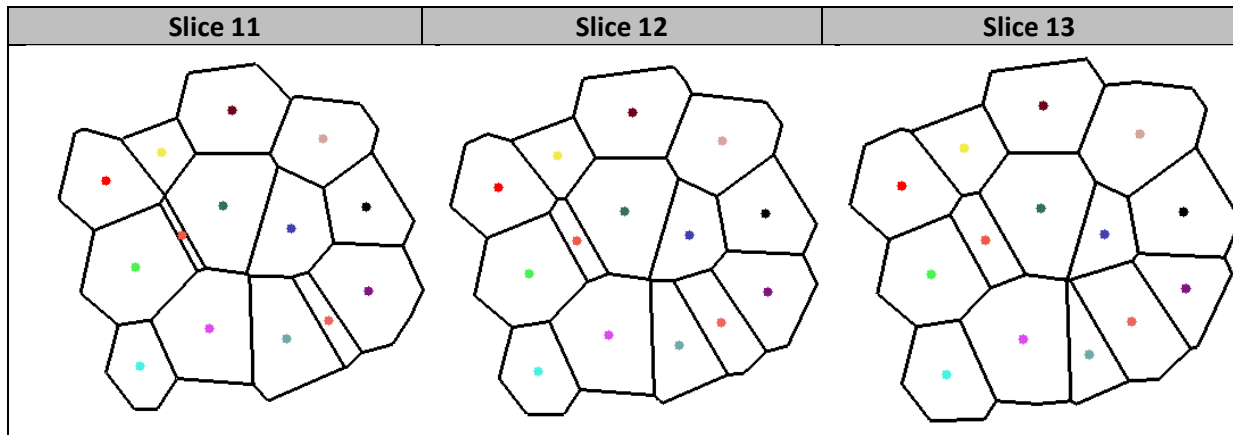


Figure 3.4: Manually associated slice regions

With the above in mind, the following displacement vector fields between slice 11 to 12 and slice 12 to 13 are obtained (Figure 3.14).

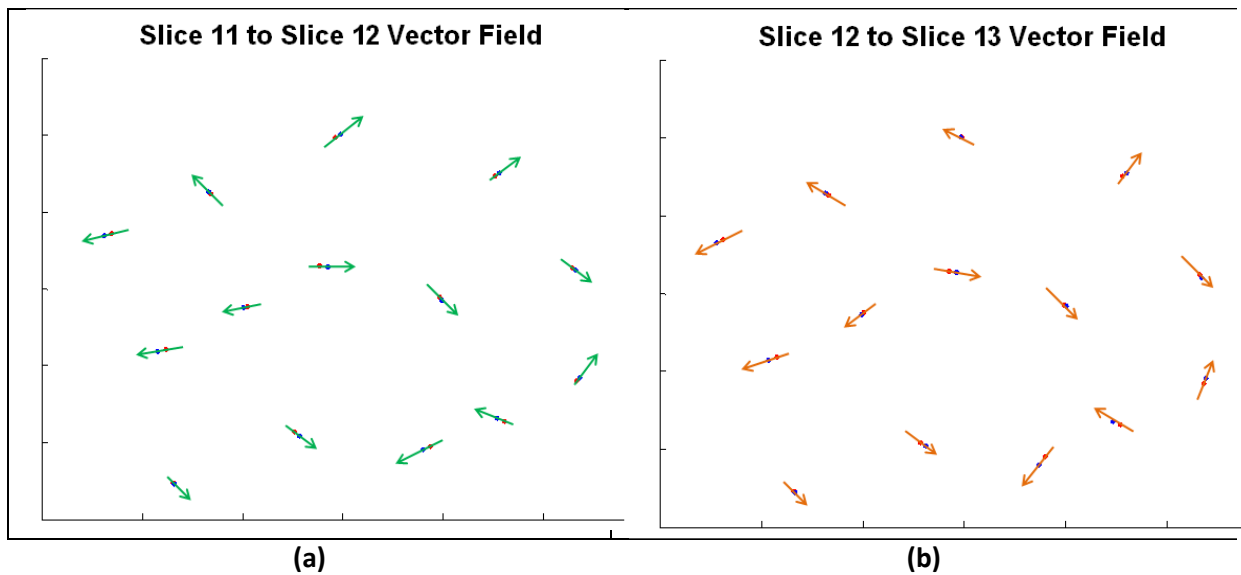


Figure 3.5: Displacement vector fields

Taking into account the exaggerated direction vectors (extending from red points to blue points), no single vector direction describes the overall displacement of contours in a slice. Nonetheless, if the vector fields in Figure 3.14 are superimposed, Figure 3.15 results.

Superimposed Vector Fields

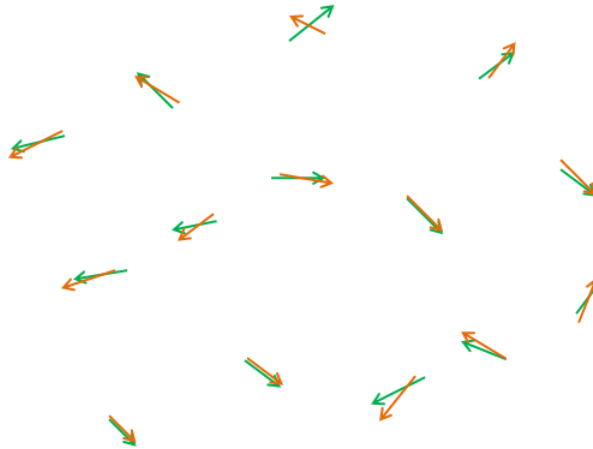


Figure 3.6: Superimposed vector fields

Figure 3.15 suggests that corresponding pairs in a displacement vector field relating three adjacent slices have a large amount of similarity. In fact, directionality of each pair of vectors (with the exception of the top-most pair) is comparable. As such, displacement of contours maintains a measure of directional continuity from one slice to the next. Development of a metric measuring displacement vector similarity may prove useful for the CGA.

Metric Definition: Bending

Figure 3.16 depicts the general case to be considered. Of three arbitrary regions on adjacent slices, vectors are drawn from region A to region B (V_1) and from region B to region C (V_2) respectively.

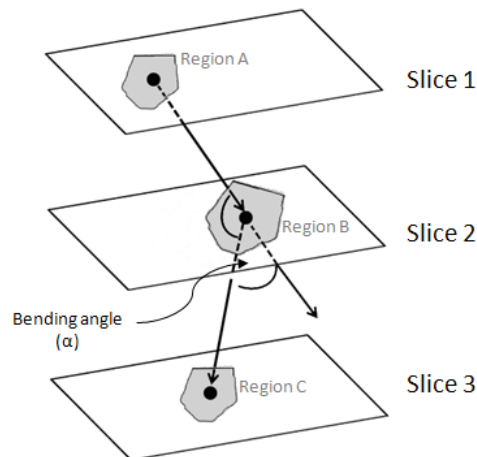


Figure 3.7: Bending visualization

These vectors naturally connect in a head-to-tail fashion, and have a bending angle (α) between them. This angle is taken as the smallest angle between a projection of $V1$ with $V2$, and is calculated utilizing Eq. 3.5.

$$\alpha = \sin^{-1}\left(\frac{|V1 \times V2|}{|V1||V2|}\right) \quad \text{Eq. 3.5}$$

Note that this angle varies on the interval $[0, \frac{\pi}{2}]$ radians, with bending angles close to 0 corresponding with the ideal case (i.e. no bending between $V1$ and $V2$). However, angle alone is not sufficient in representing a metric. Consider the vectors shown in Figure 3.17 on the x-y plane.

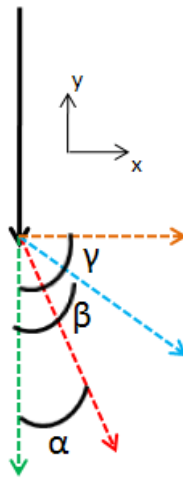


Figure 3.8: Bending angle comparison

Angles α , β and γ respectively represent angles of approximately $\frac{\pi}{6}$, $\frac{\pi}{3}$ and $\frac{\pi}{2}$. Visually, bending angles less than α are reasonable for associations, while angles above β are clearly unacceptable. Angles on the interval $[\frac{\pi}{6}, \frac{\pi}{3}]$ are more difficult to classify. There is a need to map bending angles to an equivalent value that maintains physical correspondence to “bending”, while providing a valid measure of similarity. This is required, as it provides consistent values for CGA analysis. As with overlap area, normalization to the range of $[0, 1]$ is beneficial in arriving at a final metric. In this case, metric values close to unity imply a small bending angle, and thus close similarity between regions in question.

However, normalization of bending angle is not as intuitive as that for pixel overlap. Qualitatively, the derived metric will favour angles lower than $\frac{\pi}{6}$ while those on $[\frac{\pi}{6}, \frac{\pi}{3}]$ are given marginal acceptance. Angles above $\frac{\pi}{3}$ should effectively be driven to 0, as it is difficult to imagine correct associations occurring at such high bending. With these as constraints, a suitable metric can be derived by fitting a Gaussian distribution (Eq. 3.6) with required characteristics. In fact, a number of such fits and functions may be utilized. A Gaussian has been chosen for simplicity.

$$y = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad \text{Eq. 3.6}$$

As mentioned in Section 3.2.1, Gaussian distributions are defined by a mean (μ) and standard deviation (σ). From these properties, a Gaussian distribution exhibiting constraints discussed above may be derived. Setting parameter values given in Table 3.4 and utilizing Eq. 3.6, a normalized Gaussian distribution with values between $[0, 1]$ over the range of bending angles $[0, \frac{\pi}{2}]$ is obtained. As seen in Figure 3.18, properties of this metric quantitatively satisfy previously stated constraints. The choice of property values will be explained in Section 4.1.1.

Table 3.4: Bending Gaussian fit properties

Property	Value
σ	$\frac{7\pi}{36} \text{ rad} = 35^\circ$
μ	0

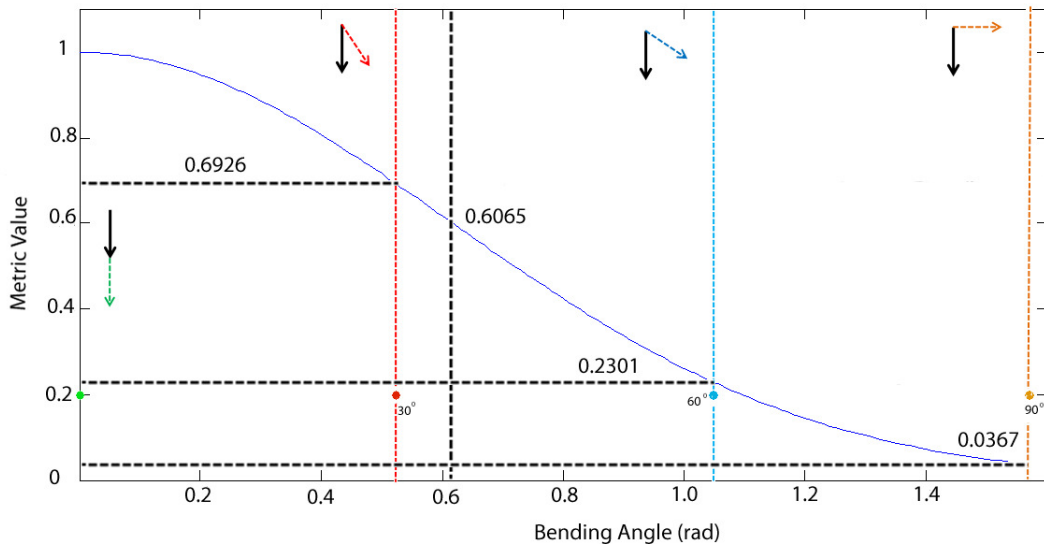


Figure 3.9: Bending metric Gaussian fit

Specifically, notice that a bending angle of $\frac{\pi}{6}$ (30°) returns a metric value of 0.6926, while a bending angle of $\frac{\pi}{3}$ (60°) gives a metric value of 0.2301. If the bending metric is interpreted as the percentage of similarity between three regions on adjacent slices, bending angles below $\frac{\pi}{6}$ correspond to similarities of greater than 70%. Bending angles between $[\frac{\pi}{6}, \frac{\pi}{3}]$ are assigned similarities of between 23% to 70%, while angles below $\frac{\pi}{3}$ are less than 23% similar. Using this fit, the calculated angle between vectors V1 and V2 are transformed into a bending metric that may be

utilized as a similarity measure in the CGA. Since this metric spans regions across three slices, it is considered a “triple metric”.

3.2.5 Coordinate Properties: Region Topology (Same Slice Neighbourhood)

Same-slice neighbourhood information is also a helpful metric for region associations. Consider that each contour in a slice inherently is related to other regions through shared boundaries. Given that contour boundaries are given as a set of pixel locations, traversal of these pixels using computational search methods allows for identification of shared boundaries between adjacent contours. Contours found to share boundaries are thus defined as neighbours. It is thus possible to define contour topology as a list of neighbouring contours within a slice. This is shown in Figure 3.19, with red vectors drawn from the centroid of the contour under consideration to those that it shares boundaries with.

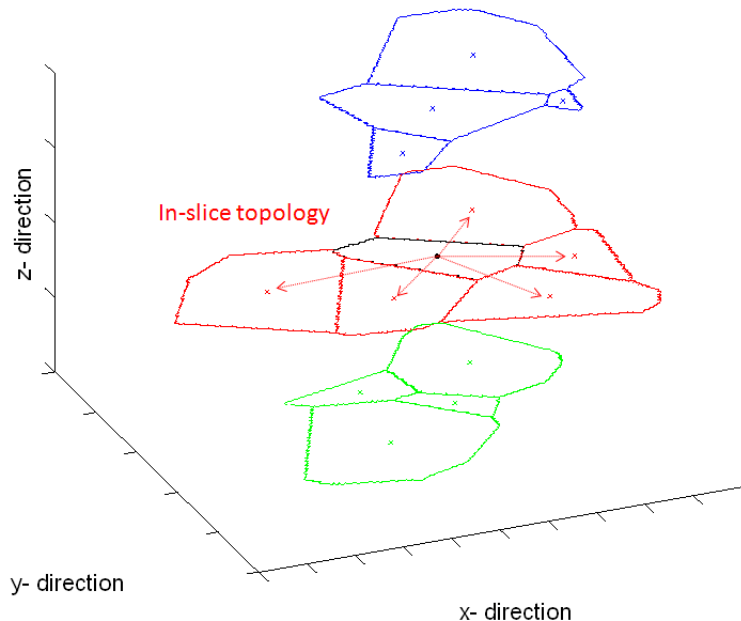


Figure 3.19: Region topology

Consequently, one can define the topology of a contour as a set of direction vectors extending from its centroid. This topological vector set provides two distinct pieces of information, the first being the number of neighbours, and the second being their relative orientation. Figure 3.20 depicts four slices of a cell mass in which topological information has been drawn for a particular contour group. Notice that as slices progress in the z-direction, there is an overall change in shape and position for each contour. However, when comparing topological profiles, topology maintains a level of consistency. Although discrepancy does exist, it is not unreasonable to suggest that topological information may be used in the CGA.

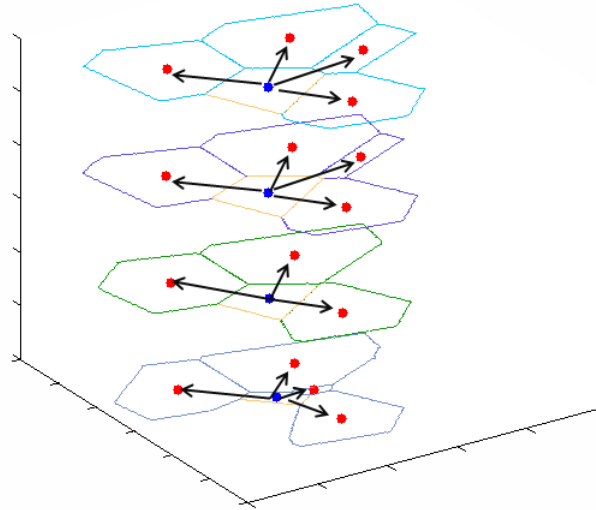


Figure 3.10: Grouping topology example

Metric Definition: Extended In-Slice Topology

The topological considerations discussed above may be used to “anchor” a contour within its immediate in-slice neighbourhood. Topological vectors extending from a contour centroid contains two distinct pieces of information: 1) number of neighbours and 2) relative orientation of neighbours. Extending this, a stronger metric is obtained by defining the topology of a single contour by its relationship with its direct neighbours, as well as the relationship of these direct neighbours with secondary neighbours (see Figure 3.21). Thus, the metric contains both direct and secondary topological information.

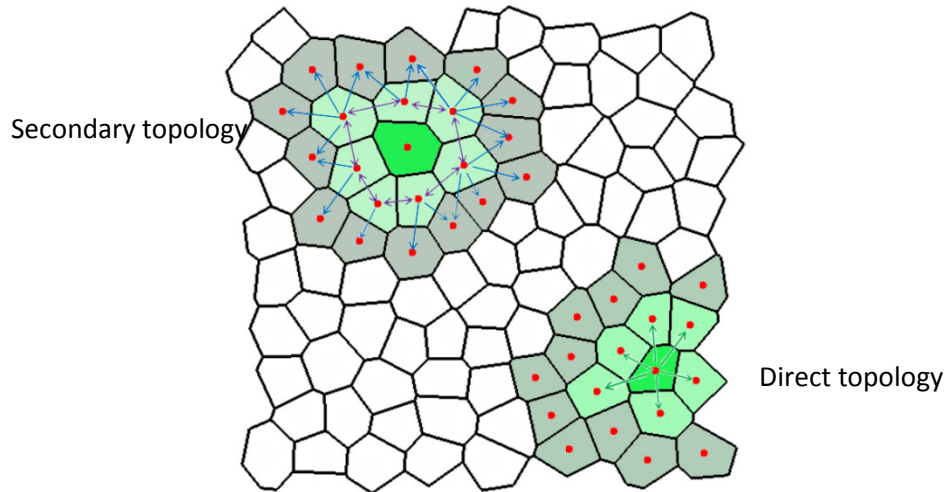


Figure 3.11: Direct and secondary topology

The conversion of topological information into a valid metric follows a heuristic approach. Consider topologies shown in Figure 3.22.

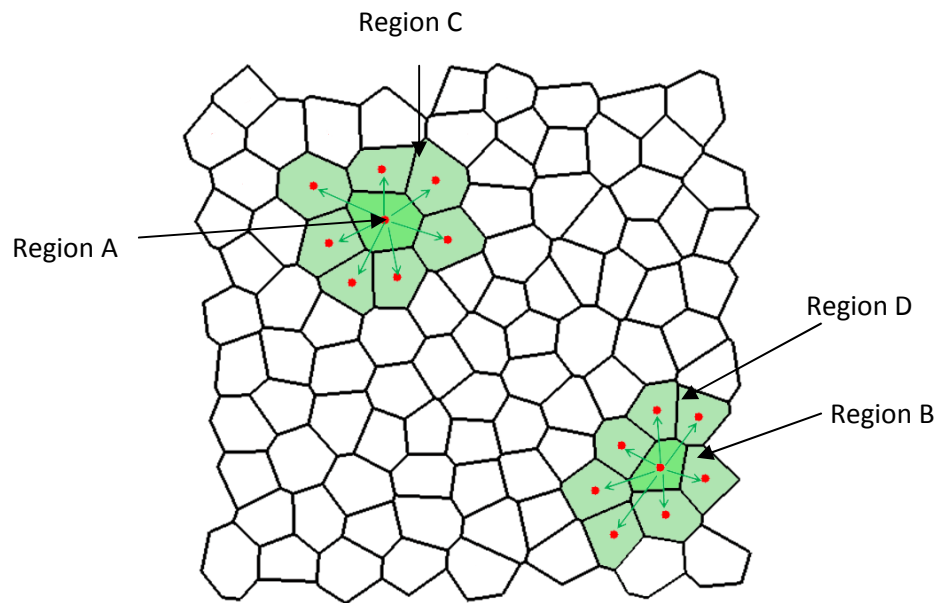


Figure 3.12: Topology comparison example

Direct neighbourhood of regions A and B are quite similar (Figure 3.23). It is also obvious that the topologies differ in number of direct neighbours (i.e. seven vs. six). Recall that the bending metric utilized Eq. 3.5 to find an angle between two vectors. In a similar fashion, the angle between pairs of topological vectors belonging to different regions can be evaluated. Subsequent conversion of the “bending” between topological vectors with respect to Eq. 3.6 once again gives a similarity metric ranging in value from $[0, 1]$.

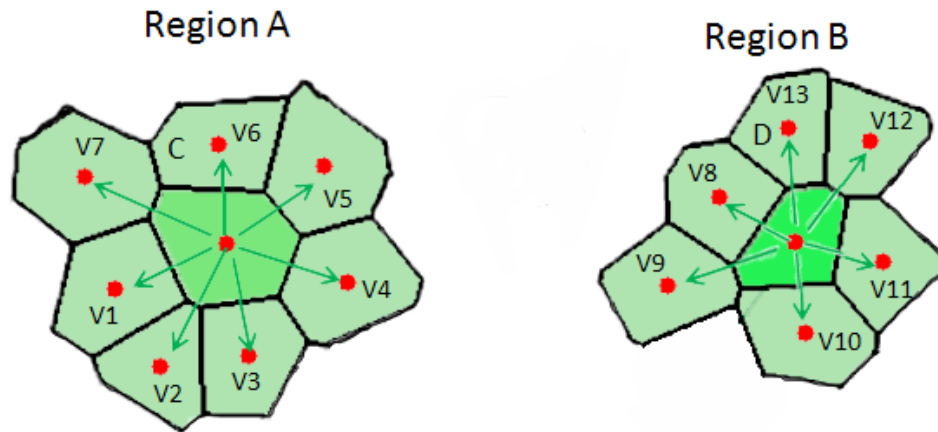


Figure 3.13: Region comparisons

Evaluation proceeds in a piecewise manner for each vector in region A with each vector in region B. For instance, V1 is compared with V8 through V12, followed by V2 with V8 through V12, and so on. This procedure results in a collection of vector similarity values given in Table 3.5.

Table 3.5: Topology comparison (Region A and B)

Similarity	V8	V9	V10	V11	V12	V13
V1	0.4371	0.9966	-0.2879	-0.6307	-0.6951	0.2985
V2	0.2623	0.5337	-0.6538	-0.2825	-0.9683	0.6939
V3	-0.4711	-0.2678	0.9896	0.3361	0.4742	-0.9752
V4	-0.9377	-0.6258	0.3253	0.9957	0.3038	-0.3108
V5	-0.3188	-0.8732	0.3742	0.4394	0.9062	-0.397
V6	0.3883	0.2902	-0.9949	-0.2934	-0.5797	0.9998
V7	0.9926	0.516	-0.3825	-0.9471	-0.2751	0.3612

Notice that this table includes negative values. These negative entries account for vectors with opposite directionality, and therefore, regardless of magnitude, do not constitute a similar topological orientation. For the case under consideration, regions A and B share five topological vectors that are above 90% similar. As part of the heuristic approach to defining topological similarity, these vectors (i.e. those pairs with similarity above 90%) are considered topological matches, and contribute to a score of 6 of 7 for direct neighbourhood match.

The next step consists of exploring each identified topological match. For each pair of vectors with 90% or greater bending similarity, corresponding contours are evaluated for topological similarity. For example, in Figure 3.22, regions C and D are evaluated for similarity. This comparison results in the following table, with an overall score of 3 of 6. The running score of the extended neighbourhood comparison is thus 8 of 13.

Table 3.6: Topology comparison (Region C and D)

Similarity	D1	D2	D3	D4	D5	D6
C1	0.5626	0.6925	-0.2723	-0.8292	-0.859	0.2623
C2	0.9489	0.2658	-0.4827	-0.7097	-0.2975	0.5787
C3	-0.3686	-0.4791	0.9998	0.2758	0.3682	-0.9708
C4	-0.9003	-0.3959	0.2859	0.997	0.5195	-0.3193
C5	-0.2948	-0.6465	-0.9267	0.2622	0.4919	-0.8343

By completing this step for each direct neighbour match, a final running score is obtained by summing scores for each comparison. For this example, the final score is 27 of 47, or 0.5745 (57.45%). The topological similarity metric is thus a fraction represented by the final running score. Within the context of the defined metrics, a value of 0.5745 implies that the extended topology of regions A and regions B are 57.45% similar.

Although not explicitly stated, number of neighbours has been included in the calculation of the running score. Consider a score of 6 of 7 obtained during comparison between direct neighbourhoods of regions A and B. Since region A has 7 neighbours and region B has 6, the actual number of potential matches is 6. However, the larger number is taken as the running denominator. Comparison between regions with differing neighbours will thus decrease metric value, since the denominator reflects a number of potential matches that can never truly be reached.

Note that the example provided above compares the topology of contours within the same slice. This was done for ease of visualization, as topology comparisons within the same slice provide no indication of contour associations across slices. However, the methodology described above may be used to compare topologies of contours existing on adjacent slices, thereby providing a metric for CGA analysis.

3.2.6 Metrics Summary

Three distinct metrics (Table 3.7) have been defined for use in associating contours across slices of confocal images. These metrics of pixel overlap, bending and extended neighbourhood similarity define relationships between two slices, three slices, and within a single slice respectively. These have been developed through consideration of geometric and coordinate properties, and may be interpreted as a percentage match between contours under comparison. The task of developing the CGA now lies with proper combination and utilization of these metrics.

Table 3.7: Metrics list

Metric	Derived From	Association Affect	Property
Pixel overlap	<ul style="list-style-type: none"> centroid position contour boundaries contour area 	across slices (pair-wise)	area overlap
Bending	<ul style="list-style-type: none"> centroid position 	across slices (triples)	displacement
Extended in-slice topology	<ul style="list-style-type: none"> centroid position 	within slice	topology

3.3 Contour Grouping Algorithm

The approach taken utilizes aforementioned similarity metrics as a probability of match between contours. Upon further consideration, it becomes clear that a probabilistic approach requires augmentation with an appropriate secondary algorithm. Since a probabilistic approach is guided by acceptance thresholds (e.g. regions with greater than 90% match probability are accepted), there exist unassociated regions even after successive iterations. To accommodate this and improve association results, a secondary algorithm utilizing the same metrics, but with different association rules, is required.

It was found that a suitable complement to iterative probabilistic associations was that of cost optimization. Given results obtained through probabilistic associations, a cost of association for each grouping may be obtained. Furthermore, it is possible to implement an optimization method where unassociated cells are assigned to existing groups through minimization of association cost. Iterations of this complementary algorithm allows for a more complete final result. These algorithms will be detailed in Sections 3.3.2 and 3.3.3 respectively. Section 3.3.1 provides a brief discussion on the preparation of source data. A summary of the approach taken is shown in Table 3.8.

Table 3.8: Summary of approach

Approach	Decision Rules	Result
Probabilistic Iterations	<ul style="list-style-type: none"> acceptance thresholds 	<ul style="list-style-type: none"> strong associations made
Greedy Cost Optimization	<ul style="list-style-type: none"> minimization of association cost 	<ul style="list-style-type: none"> optimal assignment of remaining regions

3.3.1 Data Preparation

Figure 3.24 depicts the general situation provided by confocal microscope imaging.

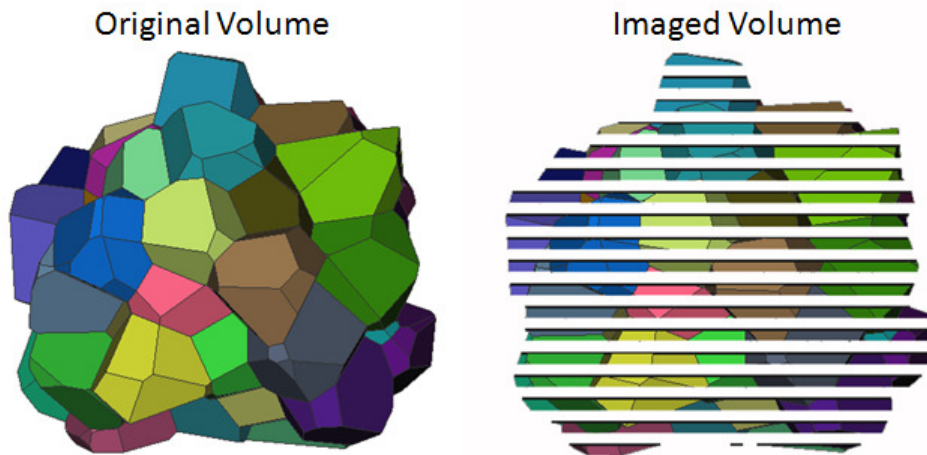


Figure 3.14: Sample volume and image stack

The original cell mass is sliced into individual image planes, resulting in a data representation with gaps. The image volume contains image information at discrete intervals along the slice axis. In this sense, the CGA works with an incomplete description of the situation at hand. Nonetheless, it is assumed that information contained in image volumes is sufficient in deriving an accurate contour grouping algorithm. Following suitable image processing, contours and centroids may be analyzed to obtain metrics described throughout Section 3.2.

Note that it is preferable to have metrics available for utilization at all times during execution of the CGA. As such, consideration needs to be made with respect to the storage of each metric. Previously, mention was made of a list consisting of the 4-nearest adjacent slice neighbours for each region. This list becomes extremely useful in building a database of metric values, as it restricts the size of the data structures required for metric storage. Each region will have eight pixel overlap comparisons and sixteen bending associations (Figures 3.25 and 3.26). Additionally, each region will have a single topological description (actual comparisons between regions are conducted later), resulting in a total number of 25 distinct pieces of information for each region. Thus, the task of the CGA is to sort through all possible combinations of calculated metrics to find reasonable cell groupings.

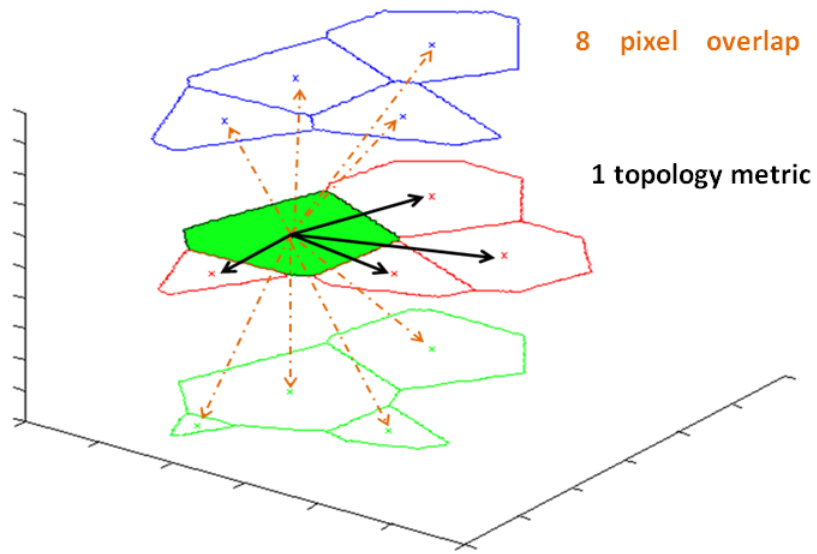


Figure 3.15: Pixel overlap associations

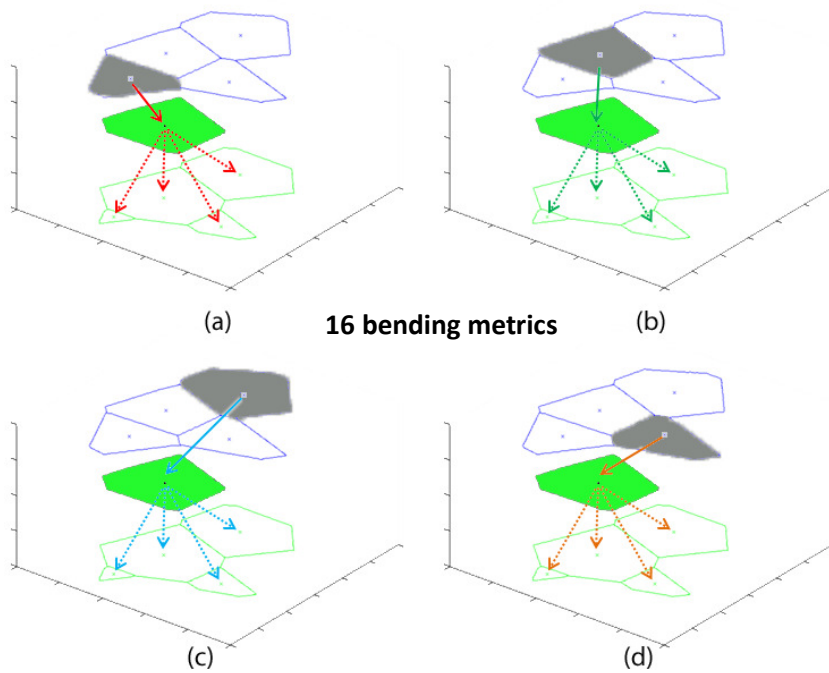


Figure 3.16: Bending associations

3.3.2 Probabilistic Iterative Associations

Triplets

The probabilistic portion of the CGA analyzes metrics for similarity values. The reader is reminded that bending is defined over three slices while pixel overlap is defined over two. In Figure 3.25, consider vectors drawn from the region under consideration to its eight neighbours. Physically, these are directional vectors connecting each neighbor with the cell in question. Combination of four “previous slice vectors” with four “next slice vectors” results in the 16 vectors portrayed in Figure 3.26.

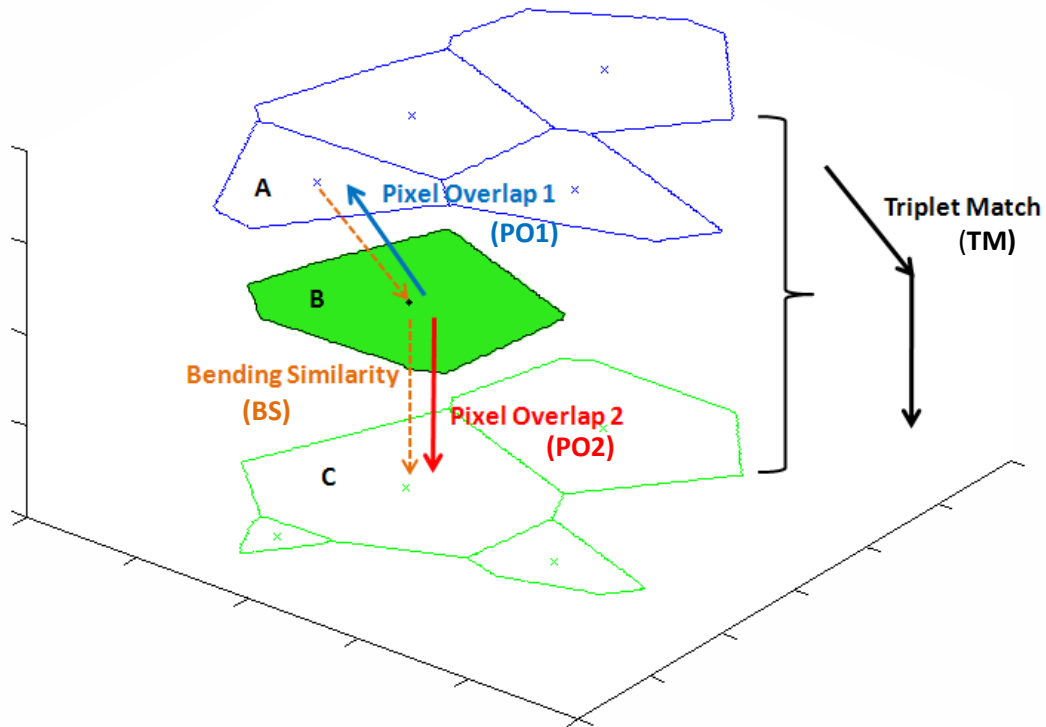


Figure 3.17: Triplet description

Considering the physical relationship between these vector sets, each bending metric may be associated with two pixel overlap metrics (Figure 3.27). Furthermore, they may be combined in a composite metric (TM). This metric is referred to as the “triplet match” between three adjacent contours, and is found with Eq. 3.7.

$$TM = BS \times PO1 \times PO2 \quad \text{Eq. 3.7}$$

Since pixel overlap and bending metrics are interpreted as probabilities of match, Eq. 3.7 may be restated as a joint probability between the three events:

$$P(TM) = P(BS \cap PO1 \cap PO2) = P(BS) \cdot P(PO1) \cdot P(PO2) \quad \text{Eq. 3.8}$$

Hence, the metrics are combined to produce a single probabilistic measure describing whether a given triplet (A, B, C) represents a valid association.

Conditional Triplets Extension

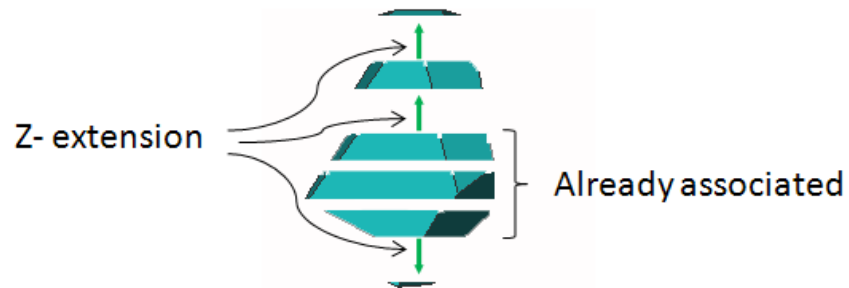


Figure 3.18: Z-extension diagram

With probabilistic triplets defined, it is only natural to borrow further from the field of probability theory. Assuming a particular triplet (TM) is accepted as truth (i.e. it is accepted as a valid association), then $P(TM) = 1$. Clearly, from Eq. 3.8, this implies that $P(BS) = P(PO1) = P(PO2) = 1$. From this assumption, slices above and below the current triplet are examined for conditional associations. This is depicted in Figure 3.29 for a single neighbour in the previous slice to an accepted triplet. Note that $P(PO1)$ and $P(PO2)$ have been dropped from Eq. 3.9 and 3.10 since they are assumed to equal 1.

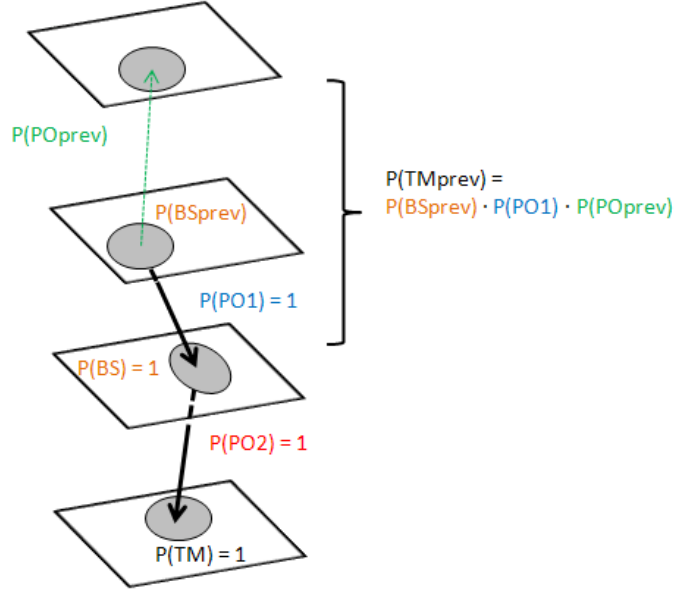


Figure 3.29: Conditional triplet diagram

The following definition for conditional triplet matches is obtained:

$$P(TM_{prev}|TM) = P(BS_{prev}) \cdot P(PO_{prev}) = P(BS_{prev} \cap PO_{prev}) \quad \text{Eq. 3.9}$$

$$P(TM_{next}|TM) = P(BS_{next}) \cdot P(PO_{next}) = P(BS_{next} \cap PO_{next}) \quad \text{Eq. 3.10}$$

As a consequence of this conditional probability definition, a framework has been established to obtain, firstly, triplets with strong probability of match and secondly, extension in the z- direction based on acceptance of strong triplets as truth. These steps are accomplished by setting a triplet threshold defining the minimum triplet match probability accepted as truth. This threshold is cross-referenced with all possible triplets, creating a list of accepted triplets. In practice, these triplets may be extended in the z- direction utilizing conditional probabilities and a modified threshold, as defined in Eq. 3.9 and 3.10. However, this method ignores topology information and is not followed. Rather, a topology check is first implemented.

Topology Check

The triplets mentioned above combine pixel overlap and bending metrics into a single probabilistic measure. The final step is to incorporate topological information such that uniform interpretation of the defined metrics may be obtained. Since topological similarity is defined as a running score of matching vectors in the extended neighbourhood of two particular regions, it is not calculated until after strong triplets are found. In this way, unnecessary computation is avoided, as only highly likely associations are evaluated for topological match.

The topology check (Figure 3.30) thus serves as a secondary test to determine triplet (R1, R2, R3) associations. In Figure 3.34, the pairings (R1, R2) and (R2, R3) are evaluated for extended topology similarity individually (P(TS1) and P(TS2) respectively). The triplet is only accepted as truth if both P(TS1) and P(TS2) are above a predefined topology threshold. If a triplet passes the topology check in addition to the initial triplet threshold, conditional Z- extension may be conducted.

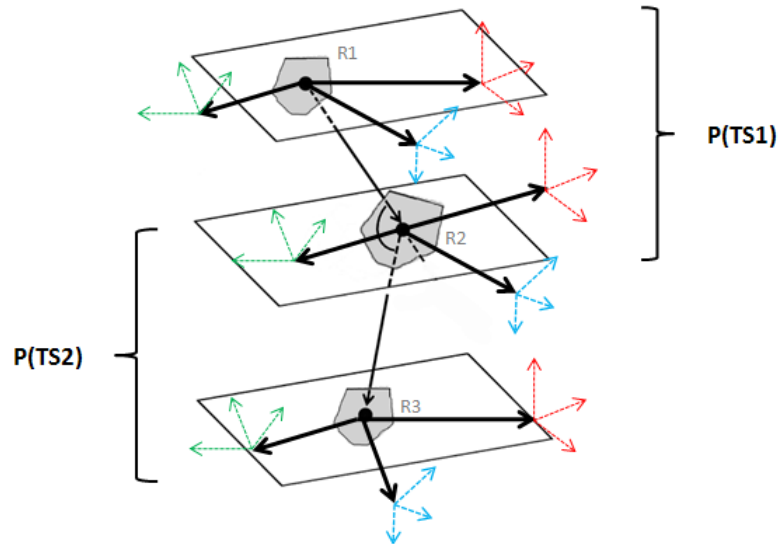


Figure 3.19: Topology check diagram

Since initial triplets are checked topologically, conditionally extended triplets are also individually checked. The procedure is identical, with topology checks conducted only to extensions that pass the defined Z- extension threshold.

X-Y Extension

A further step in the probabilistic association algorithm is X-Y extension. To this point, the probabilistic algorithm searches for initial triplets that are very strong in pixel overlap, bending and topological similarity. Conditional extension in Z grows these strong triplets, and results in a set of associated contours that span a number of slices. Since the probabilistic approach utilizes threshold levels to determine triplet associations, it is assumed that not all true contour groupings will be represented by a strong triplet. That is, a cell mass may not be sliced such that strong triplets are found for each cell.

By adjusting threshold values, it is typically possible to find at least one strong triplet. This allows Z-extension and subsequent X-Y extension. X-Y extension follows from Z- extension in that it assumes previously associated regions are representative of ground truth. Referring to Figure 3.31, a triplet (coloured in blue) has been identified and taken as truth. Its neighbor regions are evaluated for triplet similarity and, if an X-Y triplet threshold is met, also evaluated for topological similarity.

The neighbor triplet is thus accepted only if it passes both the X-Y triplet and topology thresholds respectively. Note that the X-Y triplet threshold is set lower than the original triplet threshold.

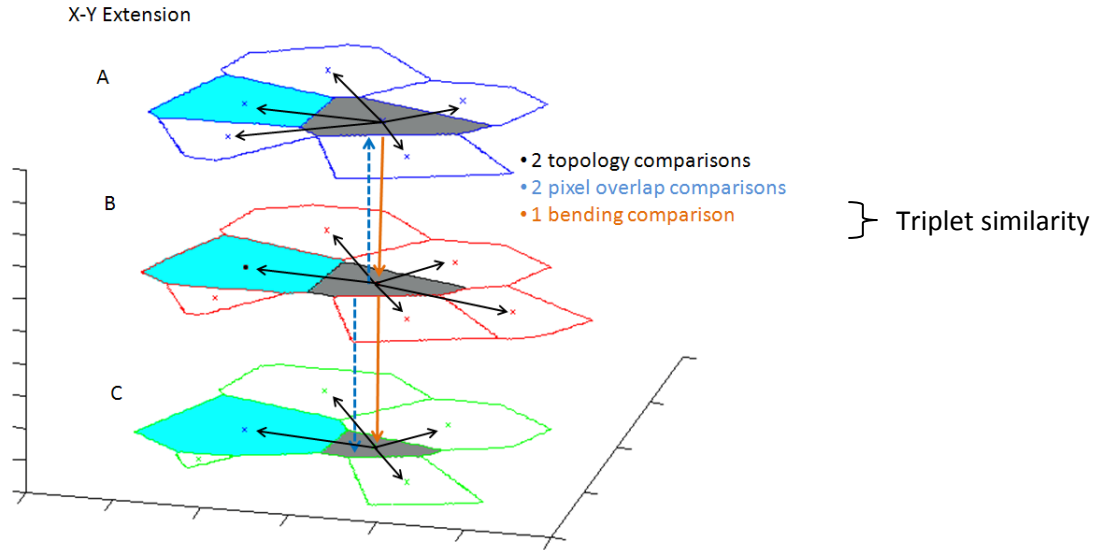


Figure 3.20: X-Y extension diagram

Summary

Algorithm steps for the probabilistic association algorithm are provided in Table 3.9. Parameters required for the algorithm are shown in Table 3.10.

Table 3.9: Summary of probabilistic association algorithm

Algorithm Steps
1. Build triplets
2. Triplet Topology Check
3. Z- Extension
4. Triplet Topology Check
5. X-Y Extension
6. Triplet Topology Check
7. Iterate 2 thru 6

Table 3.10: List of adjustable parameters

Parameters	Effect
Triplet Threshold	<ul style="list-style-type: none">• minimum triplet similarity
Topology Threshold	<ul style="list-style-type: none">• minimum topology similarity association
Conditional Triplet Threshold	<ul style="list-style-type: none">• minimum conditional triplet similarity
Conditional Topology Threshold	<ul style="list-style-type: none">• minimum conditional topology similarity
X-Y Triplet Threshold	<ul style="list-style-type: none">• minimum X-Y extension triplet similarity

To summary, the probabilistic portion of the CGA consists of four components. The first is identification of strong initial triplet similarities (Eq. 3.8) through application of a triplet threshold. These strong triplets are subsequently extended in the Z-direction utilizing conditional triplet calculations given by Eq. 3.9 and 3.10. As an intermediate step, topological checks are conducted after identification of each initial or conditional triplet. In this way, all metrics defined in Section 3.2 are applied. Furthermore, completion of Z- extension is followed by X-Y extension, a step in which the association of regions in the Z-direction is leveraged into obtaining neighbouring contour groupings in X-Y. Iterating through Z- extensions and X-Y extensions, the CGA traverses the entire volume of image slices. An example of this process is provided in the following section.

Example 1: Flat cell-sheet

Figure 3.32 depicts a synthetic sheet of cells in three-dimensions. The geometries of regions do not change in the Z-direction and have been divided into five slices. Note that each of the 100 true cells contains strong triplets. For demonstration purposes, only three of these are accepted manually. The final result is shown in Figure 3.33, with a representative set of steps displayed in Table 3.11. In truth, this flat sheet can be associated in a single iteration. However, seeding with only three strong triplets allowed the CGA to complete multiple iterations, demonstrating its behavior. Successive iteration extensions are shown in differing colours. Note also that cost optimization (yet to be discussed) was not required.

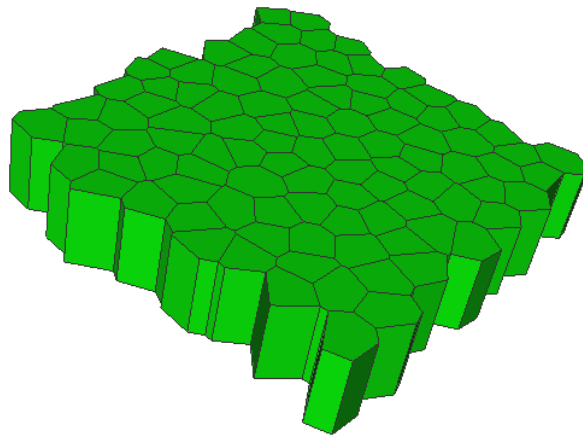


Figure 3.21: Sample cell sheet

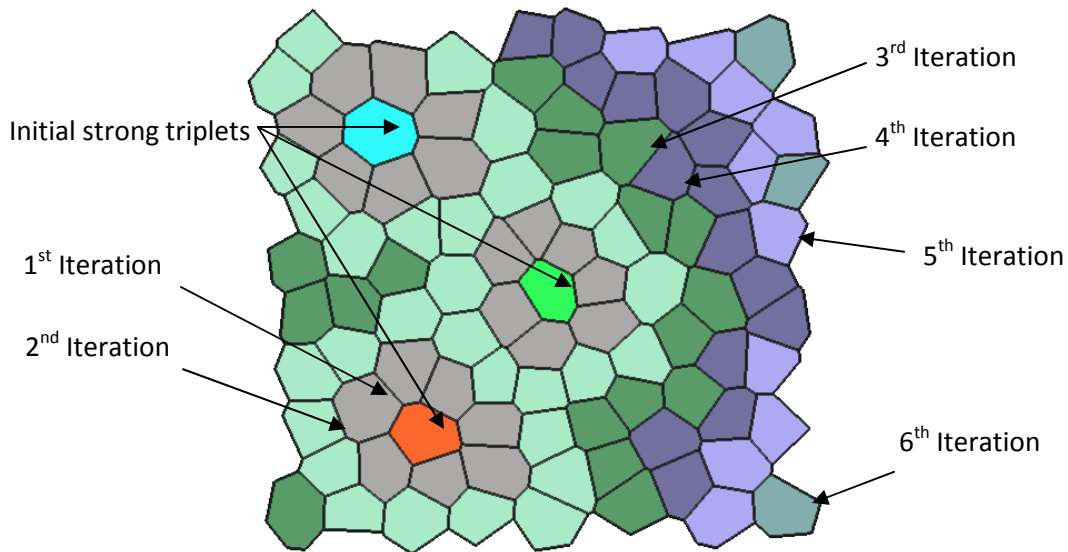


Figure 3.22: Probabilistic association result

Table 3.11: Step by step probabilistic association depiction



Example 2: Curved cell-sheet

Figure 3.34 depicts a synthetic curved sheet of cells. This case is clearly more difficult to group, as cell geometries change as slices progress in the z-direction. As with Example 1, strong triplets have been manually seeded. In Table 3.16, there remain unassociated contours. This underlines the need for a second association step, as remaining contours do not associate based upon probabilistic thresholds.

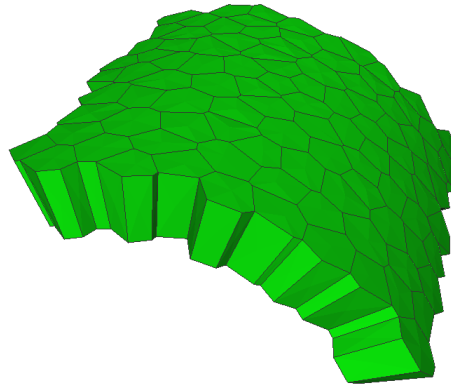


Figure 3.23: Curved cell-sheet

Table 3.12: Algorithm iterations -- curved cell-sheet

Initial Strong Triplets							
Slice 1		Slice 2		Slice 3		Slice 4	
Slice 5		Slice 6		Slice 7		Slice 8	

Table 3.13: Algorithm iterations - curved cell-sheet (cont.)

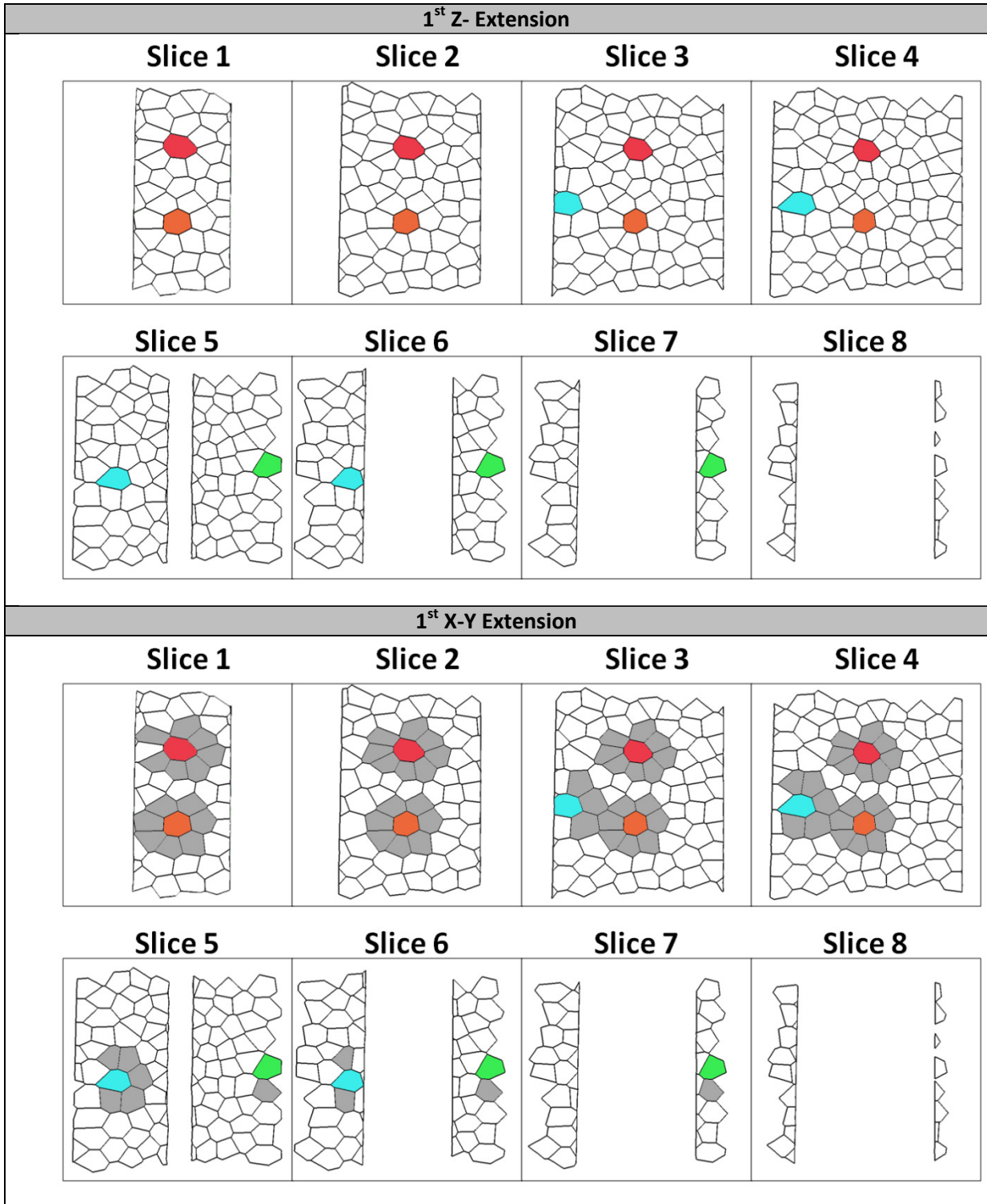


Table 3.14: Algorithm iterations - curved cell-sheet (cont.)

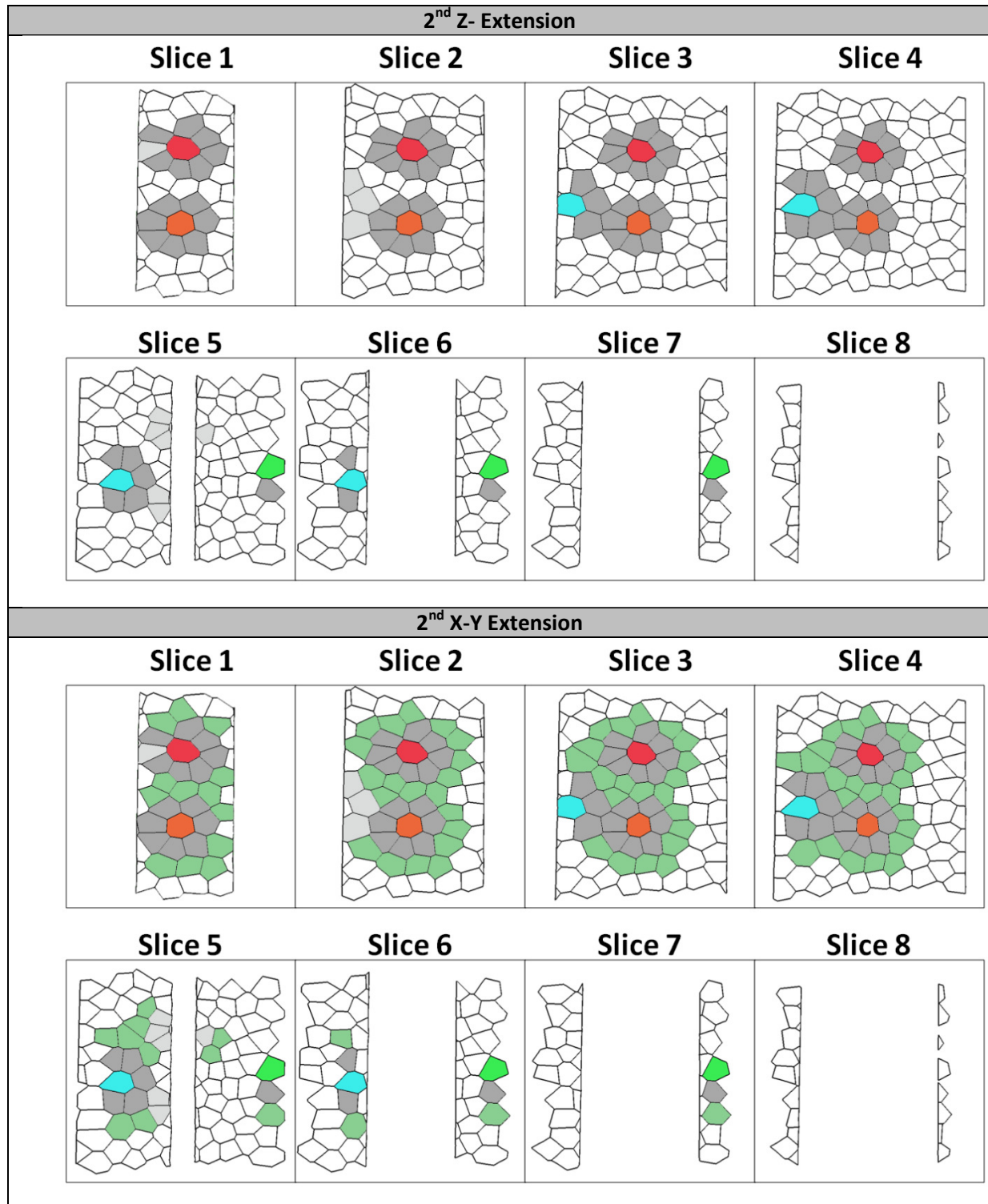


Table 3.15: Algorithm iterations - curved cell-sheet (cont.)

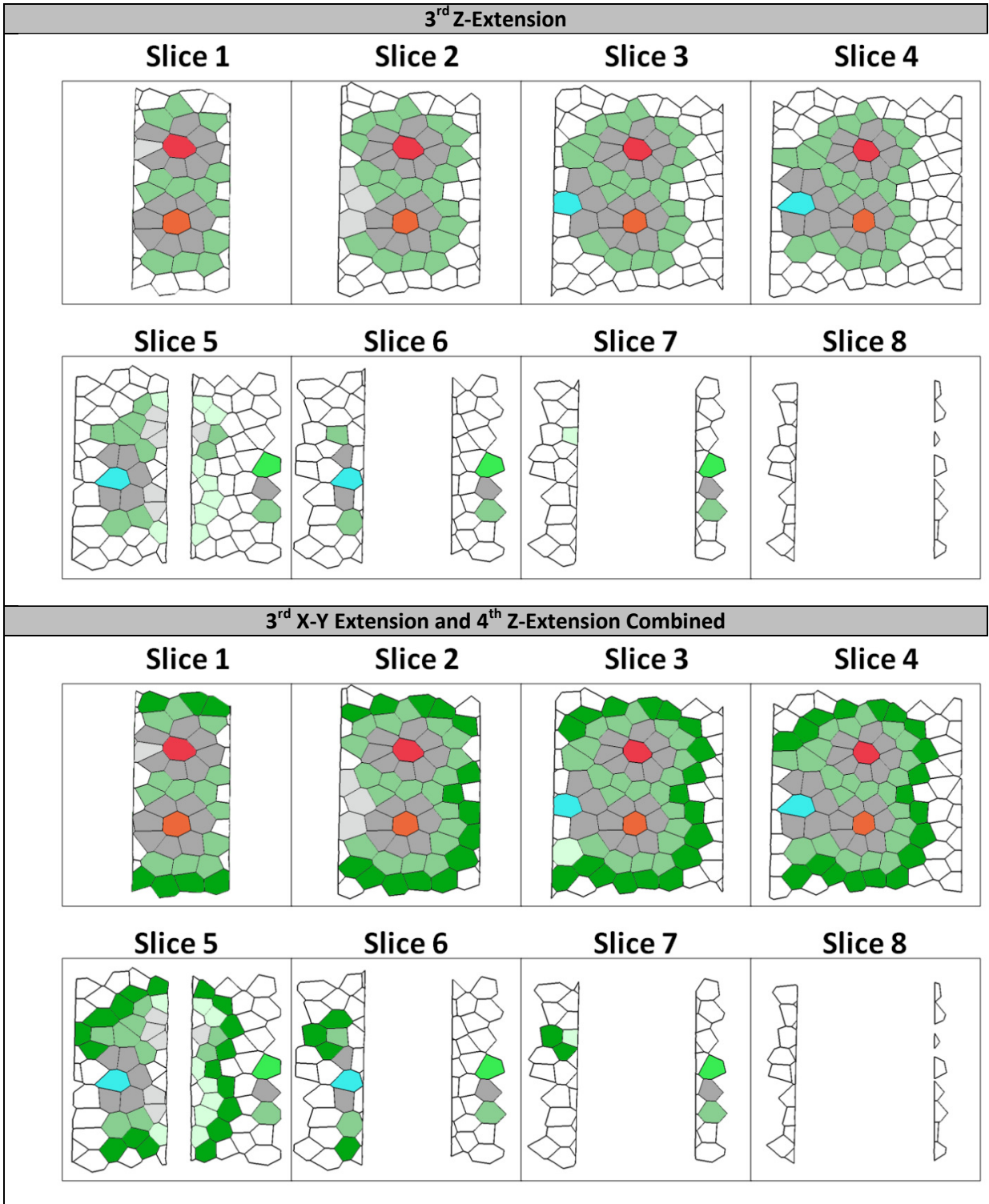
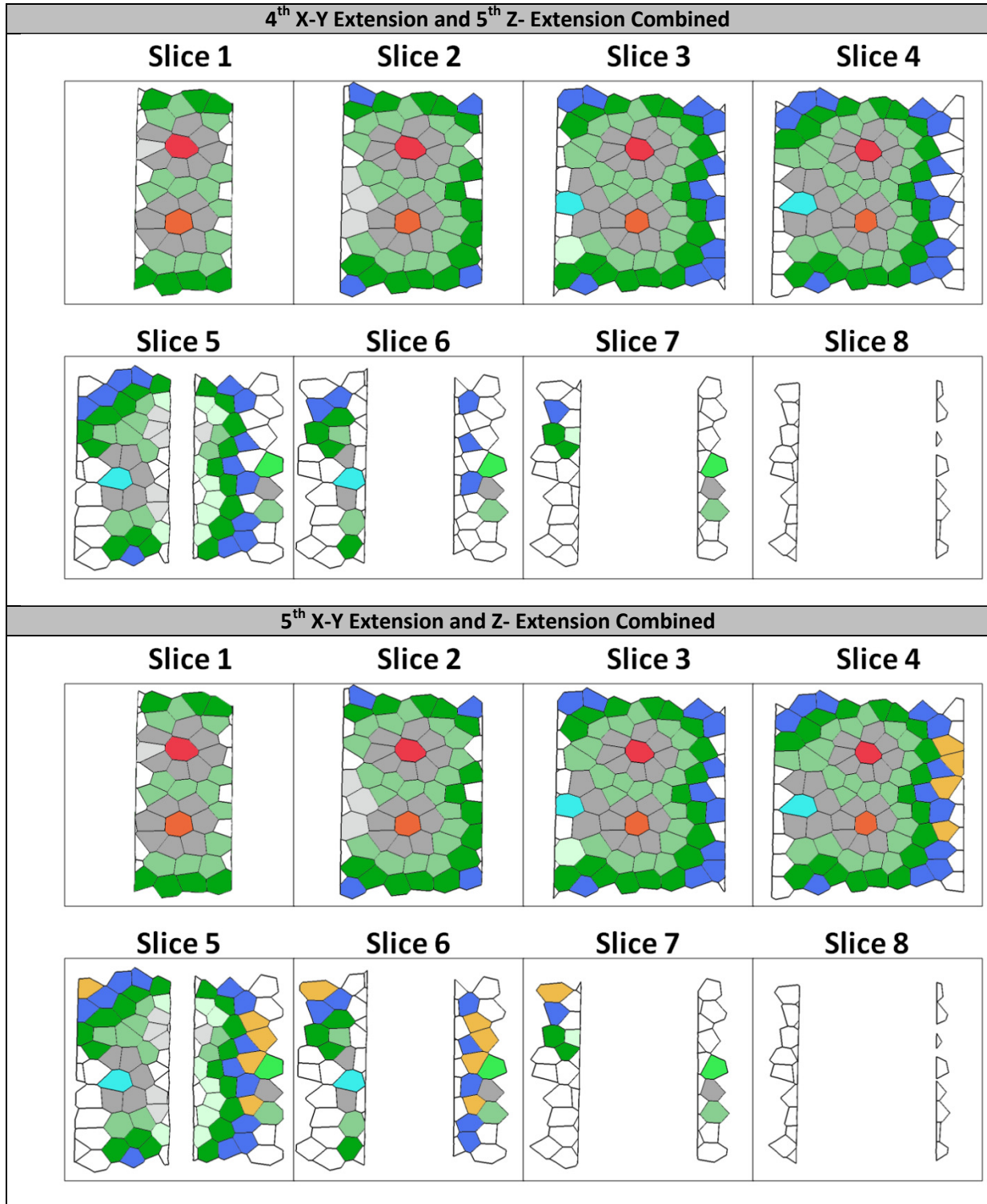


Table 3.16: Algorithm iterations - curved cell-sheet (cont.)



3.3.3 Greedy Cost Association

As a secondary association step, a greedy cost association algorithm is utilized to assign remaining contours not associated by the probabilistic approach. That is, those contours that failed to associate based on various probabilistic thresholds are associated utilizing a minimization of cost approach. Prior to implementing cost optimization, costs must first be assigned to each association metric and contour group. A “greedy” algorithm is then implemented in which unassociated regions are assigned to contour groups to which the least incremental cost occurs. Further discussion of the algorithm is provided in the following sections.

Conversion of Metrics to Cost

First, for each metric, a mapping is made between cost and similarity by defining a maximum and minimum cost (Figure 3.35). Since each metric falls within the interval $[0, 1]$, the mapping is straightforward; $C(0) = \text{maximum cost}$ and $C(1) = \text{minimum cost}$. Explicit definition of different minimum and maximum costs for various metrics, determines their relative importance (Section 4.1.3).

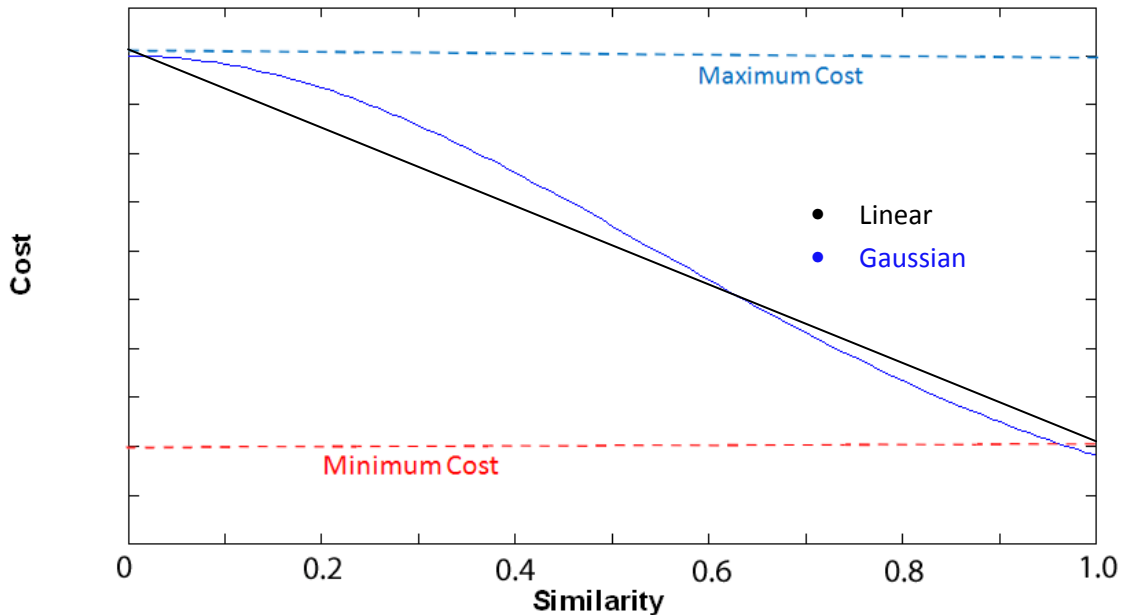


Figure 3.24: Metric conversion to cost

Exact curve fitting methodologies will not be discussed here. Needless to say, a large number of mathematical mappings exist, depending on desired properties. For comparison, a Gaussian and linear fit have been shown in Figure 3.35. The Gaussian may be more suitable, as it emphasizes the detrimental effect (i.e. large cost) of low similarity values and positive effect (i.e. low cost) of high similarity values. Table 3.17 lists the notation to be used when referring to pixel overlap, bending and topology costs.

Table 3.17: Cost notation

Cost	Association	Notation
Pixel overlap	Pairs	C_{PO}
Topology	Pairs	C_T
Bending	Triples	C_B

Additional Penalties

Based on a cost approach, one may also add additional constraints to the CGA. Given metric values previously obtained, there exist a set of independent information that describes the similarity between regions. Moreover, having completed probabilistic associations, the opportunity arises for additional constraints based on existing groupings. For example, a cost may be assigned to the sum of pixel areas for each contour in a group, thereby constraining group size. Alternatively, it may be suitable to penalize the number of slices that groupings span, also constraining group size. Clearly, a large number of constraints may be added. These are customizable to properties of particular volumetric image stacks, allowing for flexibility in the association algorithm. Currently, all penalties will be characterized by the general term $C_{penalties}$.

Cost Assignment

Assignment of cost is thus conducted by cycling through each defined grouping. For each pair of associated contours in a group, cost of association is assigned by summing the cost based on pixel overlap and topological similarity. Analogously, each triplet of associated contours is assigned cost based on its bending similarity. These are described by Eq. 3.11 and 3.12 respectively.

$$C_{pair \#a} = C_{PO} + C_T \quad \text{Eq. 3.11}$$

$$C_{triple \#b} = C_B \quad \text{Eq. 3.12}$$

The total cost of a grouping is further defined as the sum of all pair and triplet costs, along with any additional penalties (Eq. 3.13). Global cost is then defined by Eq. 3.14.

$$C_{group \#m} = \sum_{a=1}^{n-1} C_{pair \#a} + \sum_{b=1}^{n-2} C_{triple \#b} + C_{penalties \#m} \quad \text{Eq. 3.13}$$

$$C_{global} = \sum_{m=1}^{\# groups} C_{group \#m} \quad \text{Eq. 3.14}$$

Iterating through all established groupings, individual group costs and total global cost is found. Optimization will thus be conducted by minimizing global cost assuming all unassociated contours must be assigned to either an existing or new grouping. Typically, a penalty is added for

new groupings formed. The term “greedy” refers to the fact that assignments are made aggressively by identifying those associations that minimize additional group costs, and by extension, additional global cost.

Type 1 Optimization

Presumably, unassociated contours belong either to existing groupings or as yet undefined groupings. The goal of the optimization is therefore to cycle through each unassociated contour, assigning it to a neighbouring group. If no neighbouring groups exist, the contour remains unassigned, while if multiple neighbor groups exist, the contour is assigned to that which contributes a minimum total increase in group cost (Figure 3.36). This process is iteratively applied, along with Z-extension and X-Y extension, until no further assignments are made. This constitutes Type 1 optimization of the greedy optimization algorithm.

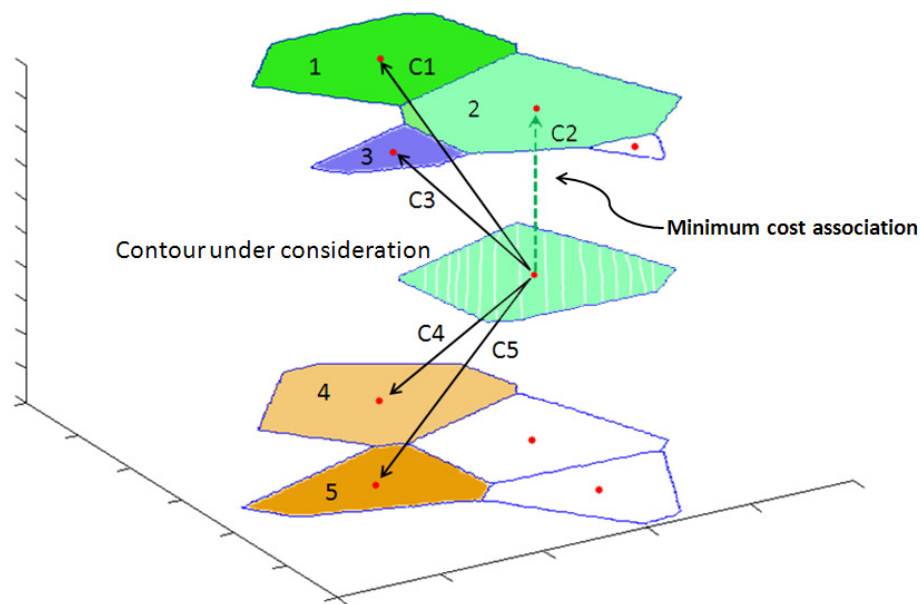


Figure 3.25: Type 1 optimization assignment

Type 2 Optimization

In some cases, even after Type 1 optimization, unassigned contours remain. Further evaluation reveals that these contours either do not neighbour any existing groupings or are very costly to associate with existing groupings. This implies that a particular contour may be better suited as a

new grouping (assuming an unassigned neighbour exists). Type 2 optimization takes this into account, systematically creating new groupings if the need arises.

3.3.4 Association Visualization

The following tables contain visualizations of associated and unassociated contours after specific iterations of the CGA. Specifically, data points shown reflect the centroid position of all two-dimensional contours in the cell mass shown in Figure 3.1. Red dots denote unassociated contours while blue points represent associated ones. No distinction has been made with regards to group membership. Rather, the figures serve to illustrate the effect of successive iterations of both the probabilistic and greedy cost approaches. There are a total of 730 contours.

Table 3.18: Visualization of iterations (region association)

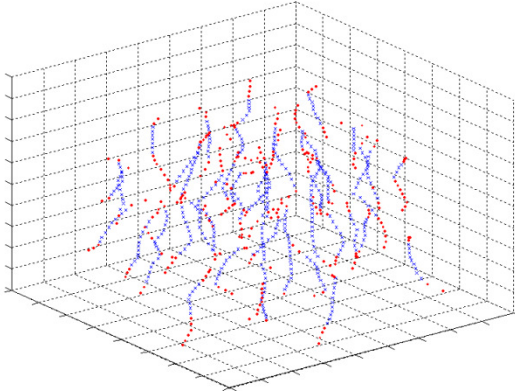
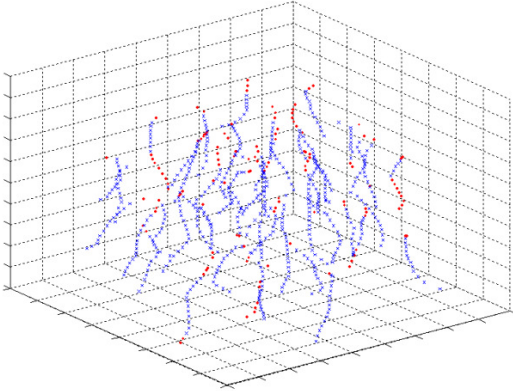
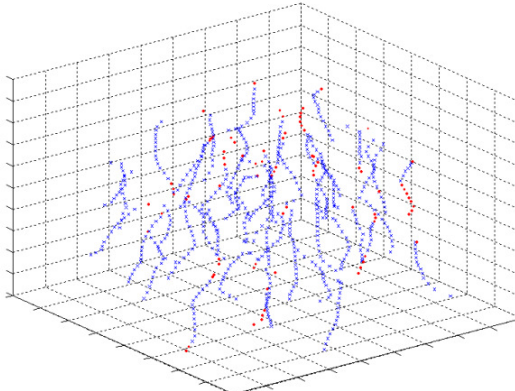
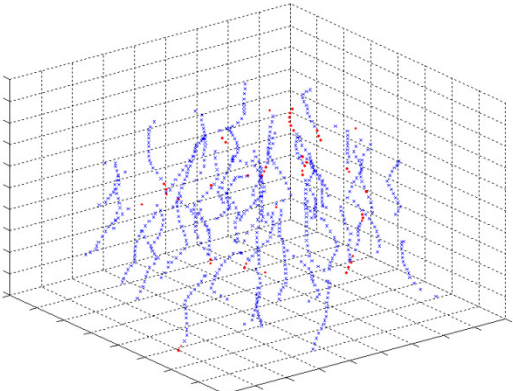
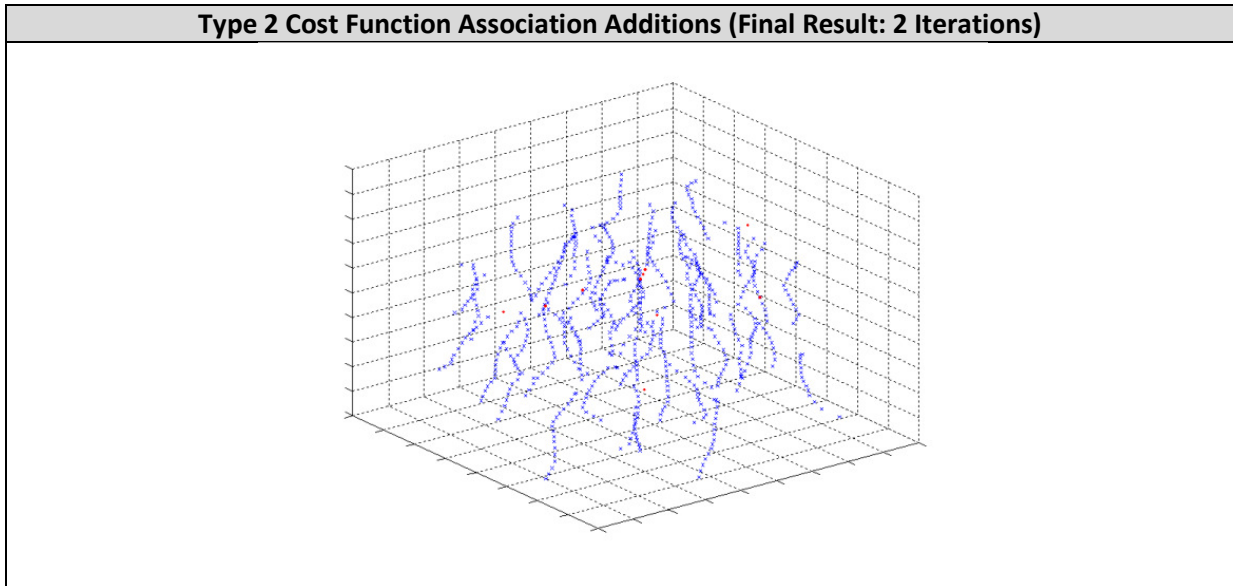
Initial Pass Probabilistic Association	Complete Probabilistic Association (2 iterations)
	
Type 1 Cost Function Association Additions (2 Iterations shown: #1 and #5)	
	

Table 3.19: Final result visualization (region association)



The probabilistic approach initializes the algorithm, assigning group membership to regions with strong similarity. Subsequent iterations of the probabilistic algorithm results in extension in the Z- direction and group branching in the X-Y plane. Iterations of the probabilistic algorithm are halted once no additional groupings or extensions are made. At this point, greedy cost association begins. Utilizing cost values converted from metrics, Type 1 optimization associates unassigned contours with neighbours on adjacent slices that afford the least incremental association cost. This is iterated until no further associations are made. Type 2 optimization then adds new groups where applicable.

From the visualizations, successive CGA iterations result in an increase of associated contours. Without regards to accuracy, this substantiates the assumptions made in developing the CGA. By combining a probabilistic approach with cost optimization, most contours within the confocal stack have been associated. There remain unassociated regions however, and these are most likely aberrant regions that require post processing, or the by-product of association errors. Nonetheless, from the initial set of 730 contours, only 11 remain. This constitutes a small percentage requiring manual intervention, and is seen as acceptable.

Chapter 4: Algorithm Details

This chapter outlines considerations regarding parameter choice, as well as typical errors encountered during analysis of CGA results. Evaluation measures to assess CGA performance are also defined. These simplify discussion of results to be presented in Chapter 5.

4.1 Parameter Choice

Parameters choice is integral to the CGA. Parameters may be selected to emphasize certain metrics, or to emphasize the utilization of probabilistic groupings over cost associations. It is conceivable that varying sets of parameter values will show increased accuracy for particular data sets. In this way, the CGA is potentially customizable to various data sets. The following sections outline the considerations taken when choosing CGA parameter values.

4.1.1 Gaussian Parameters

As discussed in Section 3.2.4, the bending metric is a value from 0 to 1. This value is obtained by mapping the range of acceptable angles, $[0, \frac{\pi}{2}]$, to the range $[0, 1]$ utilizing a suitable transformation function. For this thesis, angles were mapped to a normalized Gaussian distribution centered on $\mu = 0$ with standard deviation (σ) of $\frac{7\pi}{36}$.

An arbitrary normalized Gaussian distribution is shown in Figure 4.1. Note that only the shaded portion is considered, as this corresponds to the range of angles $[0, \frac{\pi}{2}]$. With $\mu = 0$, σ represents the angle at which bending similarity is approximately 60% (Figures 4.1 and 4.2).

That is, σ is chosen to be a value between 0 and $\frac{\pi}{2}$, where the user desires 60% similarity to lay. Choice of σ may further be guided by the shape of the Gaussian. Since the slope of the Gaussian is larger directly adjacent to σ , it is implied that an incremental change in angle close to σ contributes to a larger change in similarity value than an incremental angle change further to σ .

Referring to Figure 4.2, this is desirable, as it creates three distinctive regions in the angle-similarity mapping. The first region corresponds to angles indicating a high level of similarity, thereby requiring little differentiation in similarity values (i.e. low slope) close to 1. The second region corresponds to ambiguous angles in which similarity may or may not be high. The larger slope in this region (which contains σ), allows the CGA to implicitly define which ambiguous angles are more/less likely to represent associations. The third region corresponds to angles highly unlikely to represent associations, and therefore require little differentiation of similarity values close to 0.

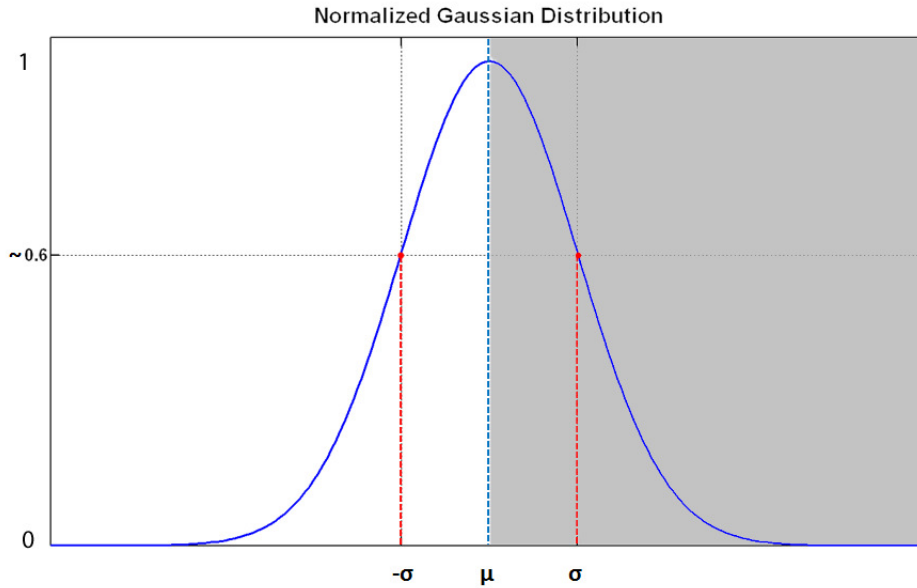


Figure 4.1: Normalized Gaussian Distribution

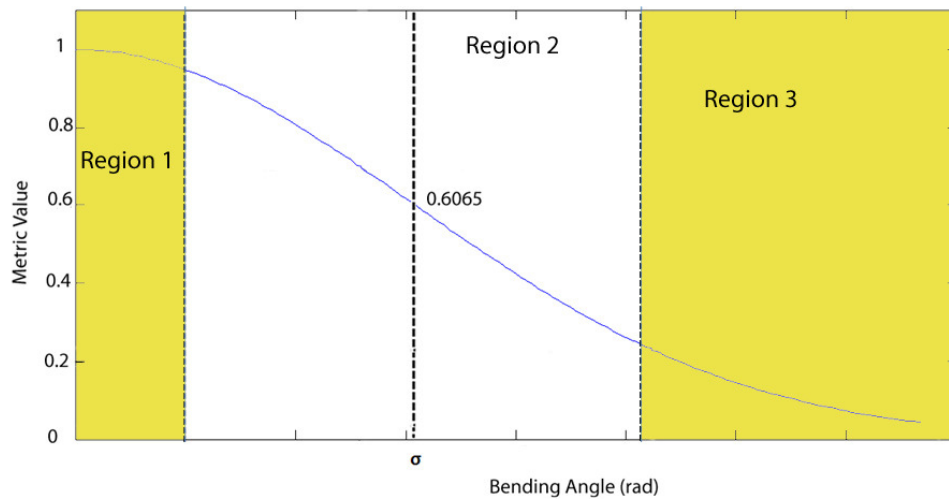


Figure 4.2: Gaussian fit regions

4.1.2 Probabilistic Thresholds

A number of thresholds are utilized in the probabilistic portion of the CGA. These thresholds represent the minimum overall similarity (as defined through methods in Chapter 3) that acceptable triplets exhibit. If a given threshold is exceeded, the triplet in question is “passed” to the next level of CGA processing. In this sense, thresholds are analogous to decision rules defining which associations are to be accepted as truth. Note that all threshold values fall in the range [0, 1].

Triplet Threshold (Thr_{tri})

Recall triplet match probability defined in Section 3.3.2:

$$P(TM) = P(BS) \cdot P(PO1) \cdot P(PO2) \quad \text{Eq. 4.1}$$

This value represents the overall similarity of a triplet based upon bending and pixel overlap metrics. Defining Thr_{tri} as a value from 0 to 1, the CGA cycles through all possible triplets and accepts only those satisfying $P(TM) \geq Thr_{tri}$. Choice of Thr_{tri} thus controls the specificity of initial triplet acceptance, with larger Thr_{tri} values indicating only highly similar triplets are accepted. The chief consideration in selecting Thr_{tri} is the type of information present in image data. For image sets where associated contours are expected to exhibit strong bending and overlap similarities, Thr_{tri} may be set to a high value (e.g. 0.9). On the other hand, if strong overlap is not expected, Thr_{tri} may be set slightly lower (e.g. 0.75).

In addition, Thr_{tri} may also be chosen to constrain the number of initial triplets accepted. That is, the user may desire acceptance of a limited number of initial triplets. In this case, the value of Thr_{tri} may be set higher than expected, causing the CGA to accept only a small number of triplets that exhibit exceptionally high similarity. Note that constraining triplet acceptance in this way effects CGA efficiency.

Topology Threshold (Thr_{top})

Section 3.3.2 defined extended in-slice topology. Utilizing this metric, accepted initial triplets are analyzed for pair-wise topological similarity. The topology check serves as a double check, ensuring that only those triplets exceeding Thr_{tri} , with topological pairs, $P(TS1) \geq Thr_{top}$ and $P(TS2) \geq Thr_{top}$, are accepted as true triplet matches.

The choice of Thr_{top} is directly related to qualities of the image set in question. In general, Thr_{top} need not be set close to 1, as this constrains associations to only those contours with near-identical topologies. Rather, a moderate value is sufficient (~ 0.5), as this disallows associations between contours with drastically differing topologies, while allowing for moderate changes caused by cell shape changes.

Conditional Triplet Threshold (Thr_{Ctri})

$$P(TM_{prev}|TM) = P(BS_{prev}) \cdot P(PO_{prev}) \quad \text{Eq. 4.2}$$

$$P(TM_{next}|TM) = P(BS_{next}) \cdot P(PO_{next}) \quad \text{Eq. 4.3}$$

Analogous to Thr_{tri} , the value of Thr_{Ctri} is set to define the minimum acceptable level of similarity for conditional triplet extension. Taking existing contour groupings, conditional triplet extension explores adjacent slice neighbours to determine whether groupings may be extended. Comparing Eq. 4.2 and 4.3 to Eq. 4.1, it is seen that conditional extensions differ from initial triplet identification

in only one term. That is, a single overlap similarity is dropped due to pre-existing groupings. As such, Thr_{Ctri} is typically set slightly higher than Thr_{tri} .

Setting Thr_{Ctri} slightly higher than Thr_{tri} acknowledges that conditional extensions accept established groupings as ground truth. From Section 3.3, this implies $P(BS) = P(PO1) = P(PO2) = 1$, thereby artificially increasing the actual correspondence between contours in a group. The increased value of Thr_{Ctri} accounts for this, while also establishing a decision rule for conditional extensions. In truth, it is difficult to determine the required difference between Thr_{tri} and Thr_{Ctri} . Nonetheless, the added flexibility of adjusting Thr_{Ctri} allows the user to customize CGA performance.

Conditional Topology Threshold (Thr_{Ctop})

The conditional topology threshold is utilized in topology checks for conditional extensions. The value of Thr_{Ctop} is set slightly higher than Thr_{top} . Once again, this is done to account for assumptions of ground truth.

X-Y Triplet Threshold (Thr_{XY})

The X-Y triplet threshold (Thr_{XY}) applies to triplet comparisons conducted during X-Y extension. Subsequent to identification of strong initial triplets, same-slice neighbours of the identified triplet are evaluated according to Eq. 4.1. If $P(TM_{XY}) \geq Thr_{XY}$, the neighbour triplet is passed on to a topology check utilizing Thr_{top} . If the neighbour triplet also passes the topology check, it is accepted as a valid strong triplet and added to the list of associated triplets for further processing.

4.1.3 Cost Function Parameters

For each metric, conversion to a suitable cost value is influenced by the choice of cost range. For instance, bending metrics may be assigned a cost range of 5 to 15, implying that the minimum cost of association for a triplet of contours with bending similarity of 100% is 5. The maximum cost of association for a triplet with bending similarity of 0% is 15. In this way, each metric is converted to cost ranges defined by the user. As discussed in Section 3.3.3, cost mappings from metrics may be accomplished using various transfer functions. For this thesis, Gaussian mappings have been utilized, similar to methodology employed for converting bending angle into a metric value.

Metric Weightings

As a consequence of metric cost ranges, relative weighting of metrics is implicitly defined. That is, within the framework of cost optimization, metrics are assigned importance based upon cost ranges assigned to each. Consider Figures 4.3 and 4.4. Figure 4.3 depicts a range of [0, 20] for topology and [5, 15] for overlap. Notice that low topology similarity (0 to ~0.2) contributes a higher cost of association than a similar overlap similarity. Additionally, high topology similarity (~0.4 to 1) contributes less cost than the minimum cost for overlap. As such, it is implied that topology is more “important” in the context of cost optimization.

Similarly, Figure 4.4 depicts the cost ranges between bending and topology. In this case, both ranges have a minimum of 0, but with differing maximums. Once again, topology is implied to be more important, since its range of values is larger than that of bending. Thus, referring to Figure 4.5, superposition of all three cost ranges indicates that metric importance is given by: 1) topology, 2) bending and 3) overlap.

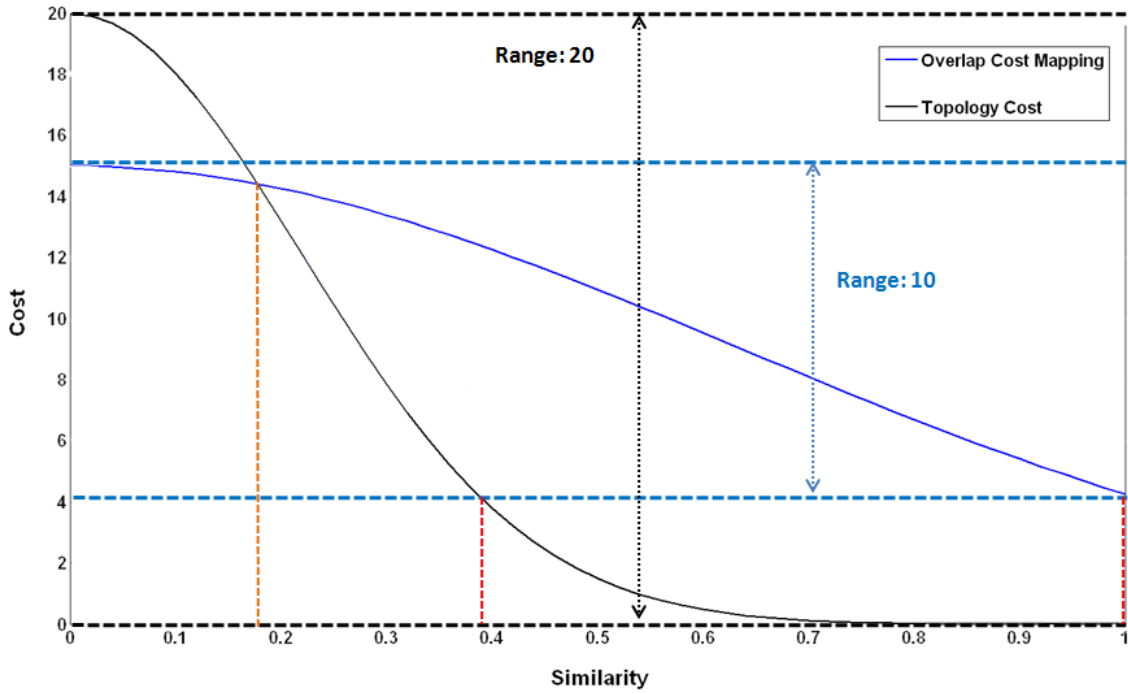


Figure 4.3: Overlap and topology cost

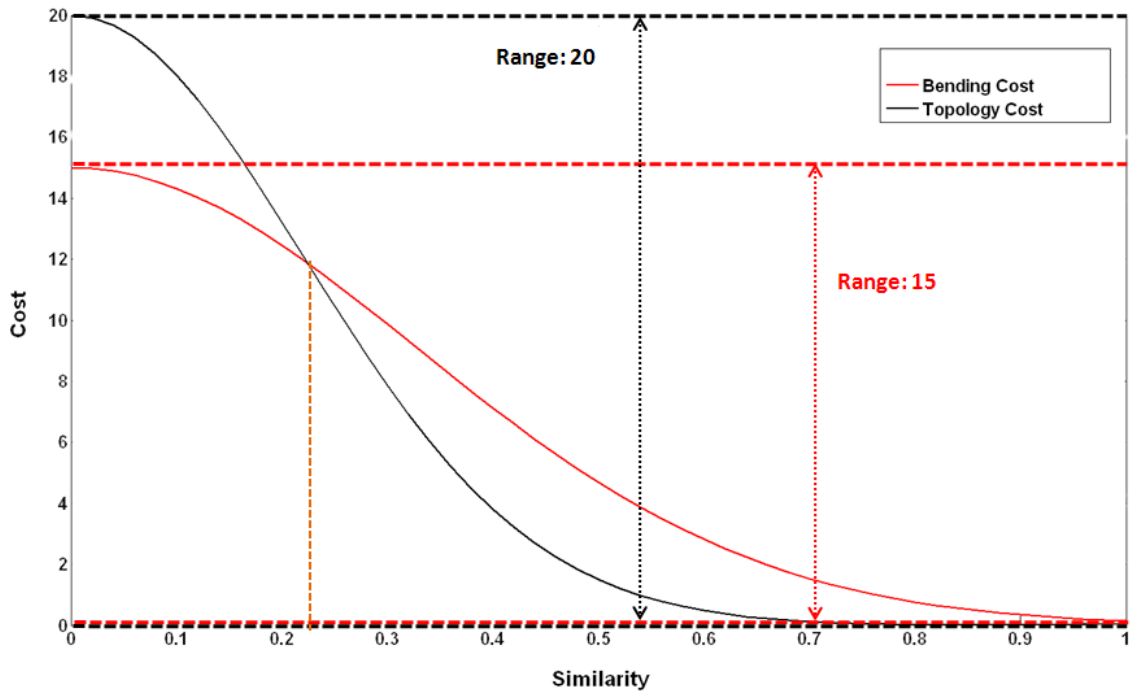


Figure 4.4: Bending and topology cost

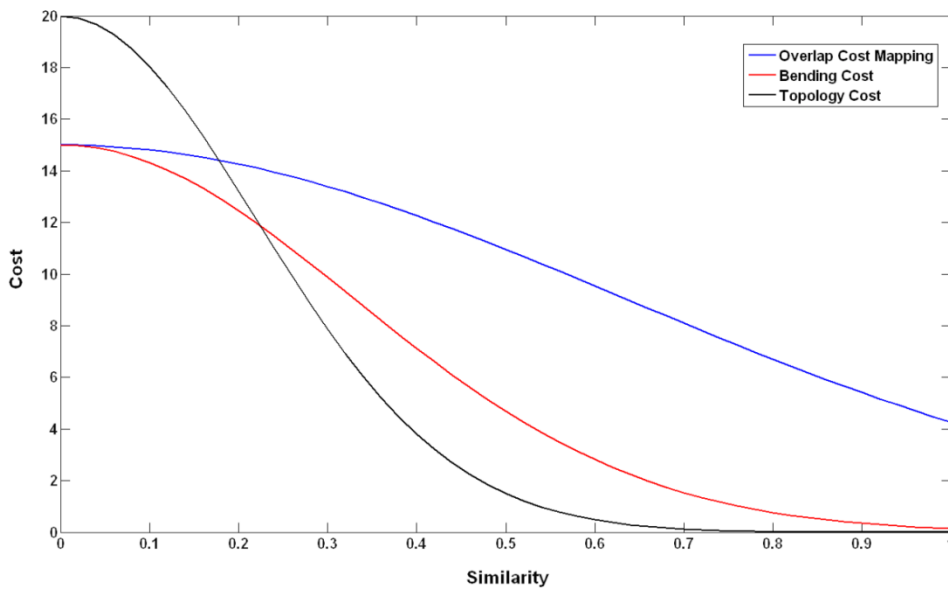


Figure 4.5: Overlap, bending and topology cost

With this in mind, the choice of metric cost ranges depends largely upon desired behavior of cost optimization. It is possible that cost ranges may be leveraged to take advantage of particular image properties. For example, overlap may be a strong determinant of associability in certain data sets. In such cases, the cost range for overlap may be tuned specifically to take advantage of such

prior information. Once again, CGA parameter provides flexibility in implementing contour associations.

Penalty Costs

The cost framework presented in Section 3.3.3 included optional penalties. Penalty costs are typically applied to incremental property changes (i.e. difference in contour areas, difference in number of contours in grouping to average, etc.), and are thus assigned incremental penalties. Specifically, a cost is added for each unit difference in the penalized quantity (Figure 4.6). As such, the penalty value for a particular quantity is defined as the rate of change of cost per unit difference.

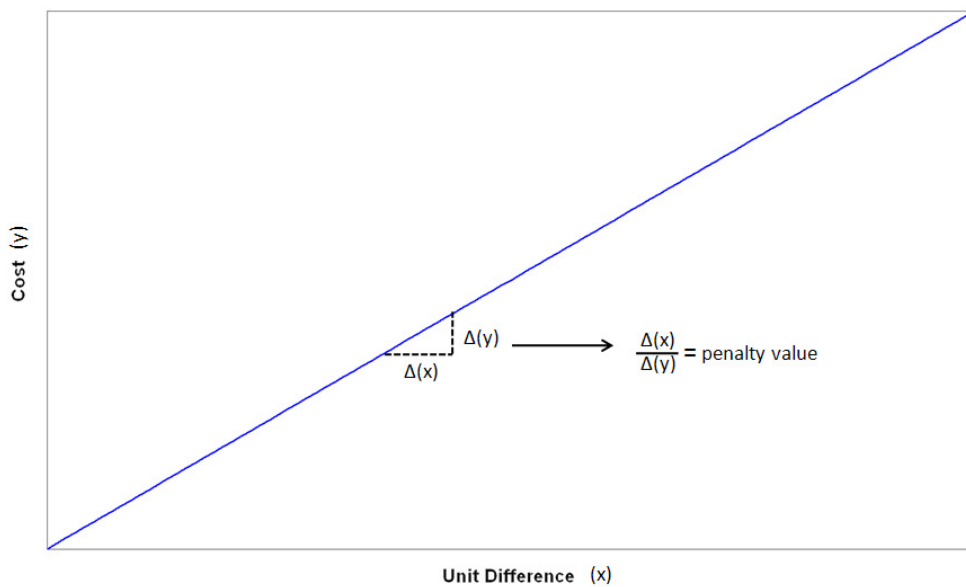


Figure 4.6: Penalty cost

Thus, the choice of penalty values depends largely upon the desired penalization effect. A large penalty cost will serve to discourage particular groupings, whereas low penalty costs will be indifferent. Penalty values must be chosen to reflect relative importance of various penalties, as well as the cost of association afforded by metric cost ranges. That is, if penalty values far outweigh costs of associations due to metrics, penalty costs will end up driving cost optimization. A balance must be struck between metric cost ranges and penalty costs to produce ideal CGA behaviour.

4.2 Error Types

This section outlines three types of errors typically encountered by the CGA. Each error type will be described in detail. Evaluation of CGA performance in subsequent sections will largely be judged by the occurrence of these specific error types. A summary is provided in Table 4.1.

Table 4.1: Error details

	Type 1	Type 2	Type 3
# true cells affected	1	2	2
# incorrect CGA groupings	2	1	2
# incorrect associations	1	1	2
Symptom	• split cell	• merged cells	• incorrect groupings

4.2.1 Single Cell Errors

Type 1 Error

Taking isolated cells similar to that depicted in Figure 4.7a, Type 1 errors occur as seen in Figure 4.7b. Of the cells shown, each correct contour grouping has been split in two. A slice-by-slice depiction is provided in Figure 4.8. Note that a single wrongly associated contour pair (slices 15 and 16) exists. Thus, Type 1 error refers to situations in which a single association error leads to the splitting of correct groupings.

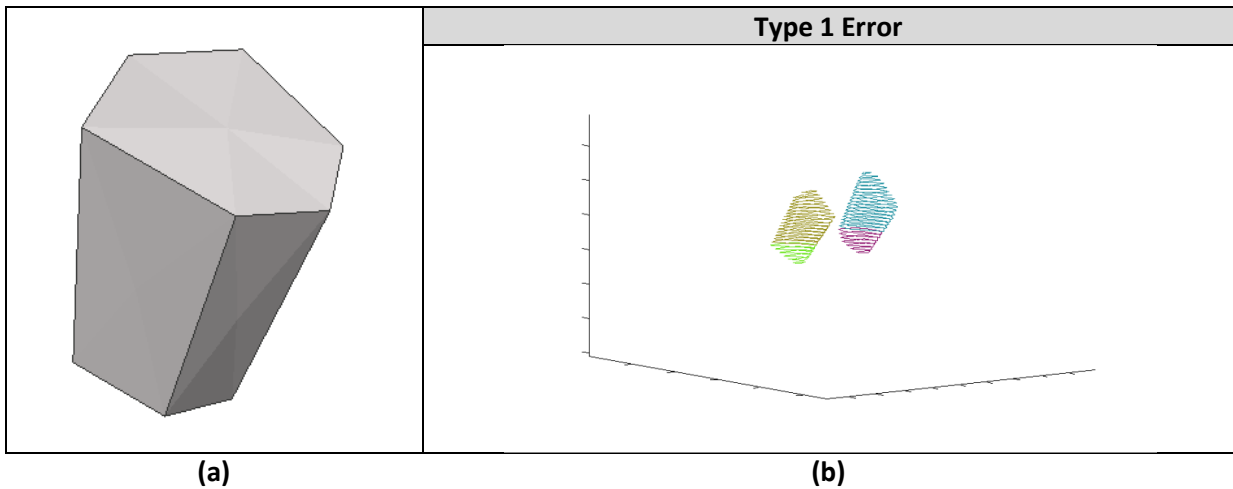


Figure 4.7: Typical Type 1 error

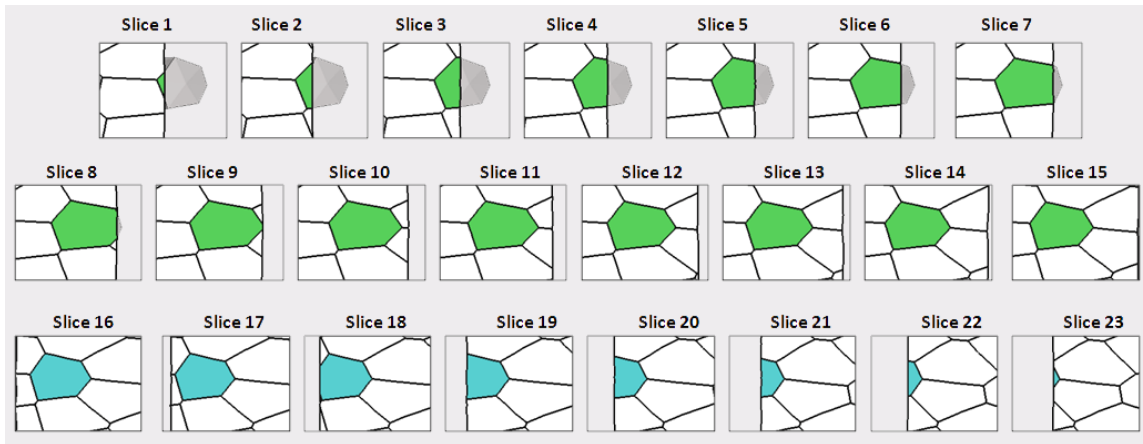


Figure 4.8: Slice-by-slice Type 1 error

If an additional association between highlighted contours in slices 15 and 16 is made, the correct grouping shown in Figure 4.9 results. Type 1 errors are therefore corrected by merging contour groups through an additional association. However, it is difficult to correct such errors within the current CGA framework, as it is not possible to automate detection of wrongly associated pairs. From the snapshots provided, source of error causing Type 1 errors is unclear. Parameters such as topology thresholds, conditional triplet thresholds or others may be the cause of mis-association. Nonetheless, the occurrence of Type 1 errors may be found manually and utilized to measure accuracy of the CGA.

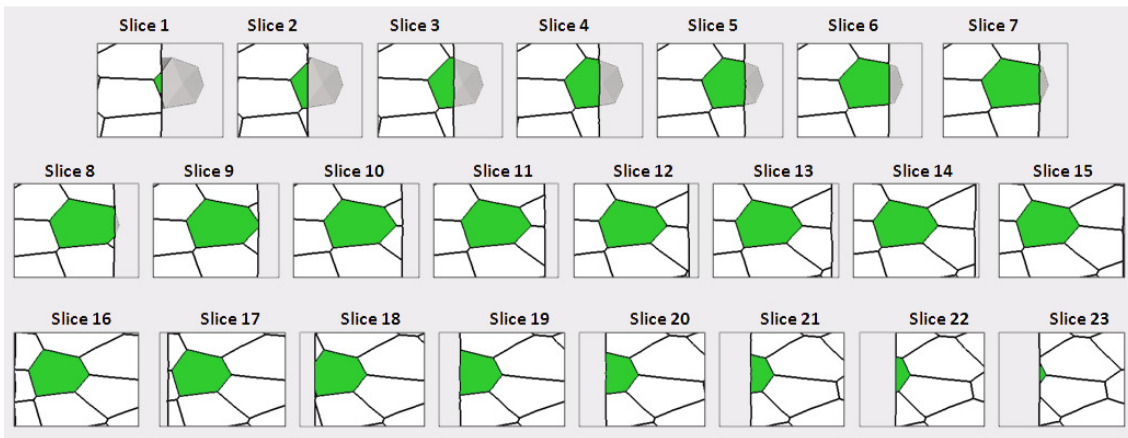


Figure 4.9: Slice-by-slice Type 1 error resolution

Furthermore, Type 1 errors may also occur in series. Although rare, multiple Type 1 errors may exist within contours belonging to the same true cell. An example is shown in Figure 4.10, in which two unique wrongly associated pairs has split the true contour group in three. With respect to the CGA, Type 1 errors in series are not much different from typical Type 1 errors. Once again, they may be caused by any parameter value, and do not provide a clear indication of what modifications may correct the error. Nonetheless, Type 1 errors in series suggest that performance of the CGA is slightly less effective than cases in which isolated Type 1 errors occur.

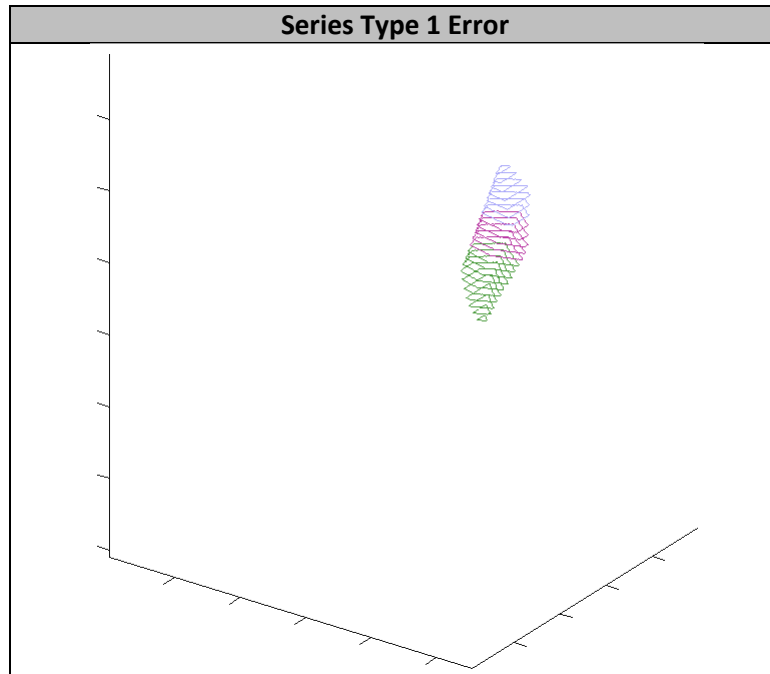


Figure 4.10: Series type 1 error

4.2.2 Adjacent Cell Errors

Type 2 Error

Given a pair of adjacent cells (Figure 4.11), Type 2 errors occur as seen in Figure 4.12a. In this case, two adjacent cells have largely been grouped into a single contour group. This group is irregularly shaped and exhibits a unique junction between cells (Figure 4.12b). Closer inspection of cell junctions reveals the primary cause of Type 2 errors is image slicing.

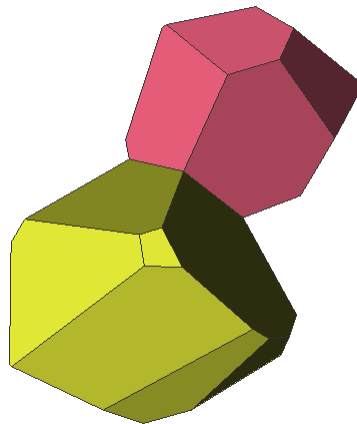


Figure 4.11: Diagram of adjacent cells

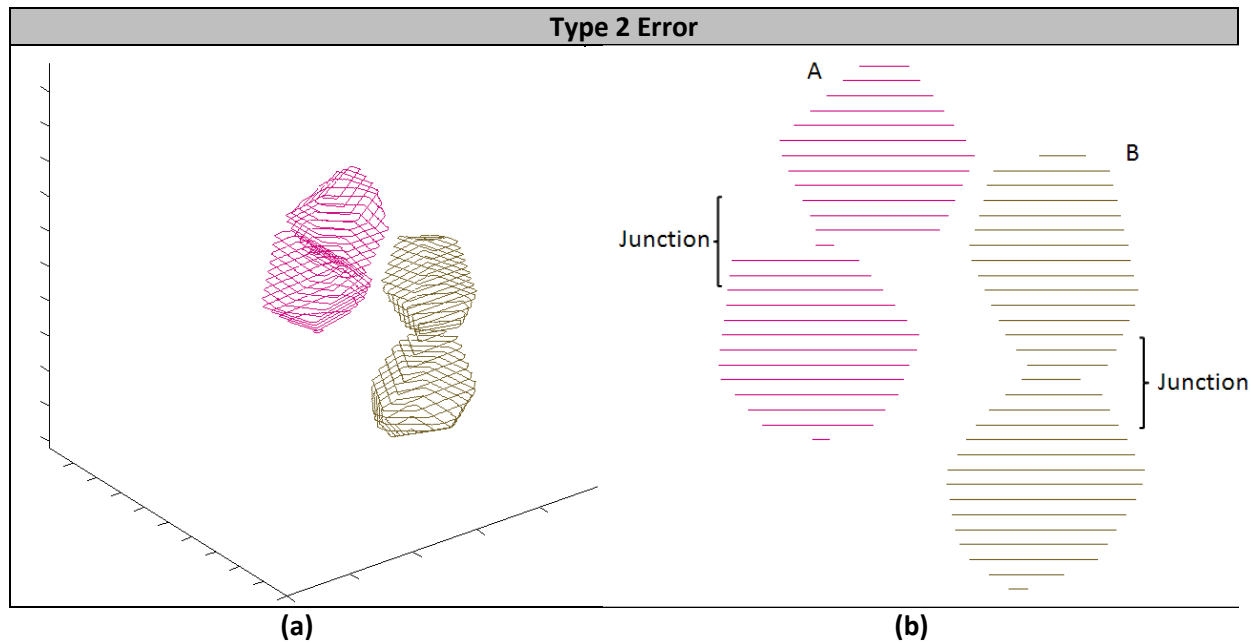


Figure 4.12: Typical Type 2 error

For example, considering Figure 4.12b, it is seen that two distinct cell shapes are associated via a small intermediate contour within the junction. It is inferred that this junction represents the region in which a cell appears to “pinch-off” and an adjacent cell begins to “grow”. On a slice-by-slice basis, two cases (Case A and B respectively) of Type 2 error are possible. The junction in Case A is shown in Figure 4.13. Contours highlighted in pink represent the CGA produced grouping, while contours highlighted in gray are unassigned by the CGA. The labels ‘1’ and ‘2’ respectively correspond to correct manual groupings.

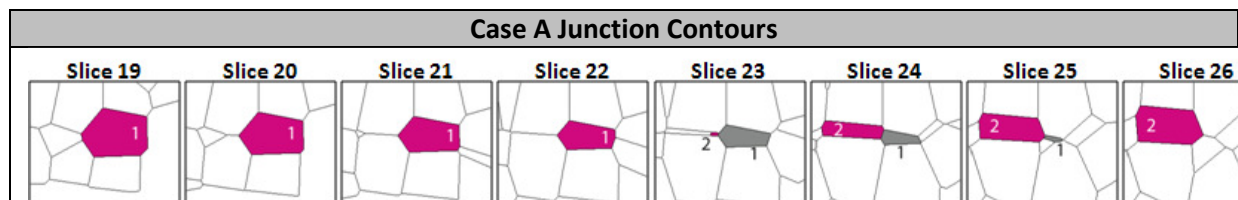


Figure 4.13: Case A Type 2 error

As seen, the CGA has grouped contours belonging to both groups ‘1’ and ‘2’ into a single grouping. Progressing from slices 19 to 22, group ‘1’ has been correctly assigned. However, in subsequent slices (during which group ‘2’ begins to “grow”), the CGA has erroneously assigned contours belonging to group ‘2’ with those belonging to group ‘1’. Additionally, three contours belonging to group ‘1’ on slices 23-25 have been left unassigned. This is characteristic of many Type 2 errors, and may be utilized to explain the cause of Type 2 errors.

With respect to the CGA, Type 2 errors may be attributed to ambiguity in similarity values between contours at the junction of adjacent cells. Specifically, in slices 22 and 23, ambiguity in similarity between the pink contour in slice 22 to contours labeled ‘1’ and ‘2’ in slice 23 has resulted

in the incorrect assignment of contours. Changes to parameter values may correct this, as similarity rules and cost evaluations may change.

Case B (Figure 4.14) represents a second class of Type 2 error. In this case, the true groups, '1' and '2' have been merged into a single CGA grouping. Similar to Case A, the junction between two true groups possess a "pinching off" and a "growing" cell. In this case however, the two cells are never seen on the same slice, and therefore result in no unassigned regions. Rather, the two true groups appear as a single merged CGA grouping.

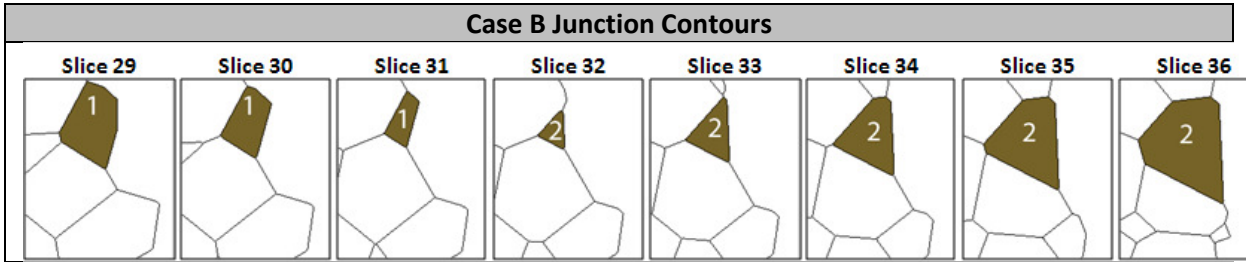


Figure 4.14: Case B Type 2 error

In either case, error is caused by a single association error causing the merging of adjacent cells. In Case A, this has occurred between slices 22 and 23, whereas in Case B, it is seen in slices 31 and 32. Similar to Type 1 errors, this may be due to any of a number of parameter values and may occasionally manifest in series (Figure 4.15). Automatic detection of Type 2 errors is difficult, and is currently not implemented in the CGA. Nonetheless, identification of Type 2 errors is useful in providing a measure of CGA accuracy.

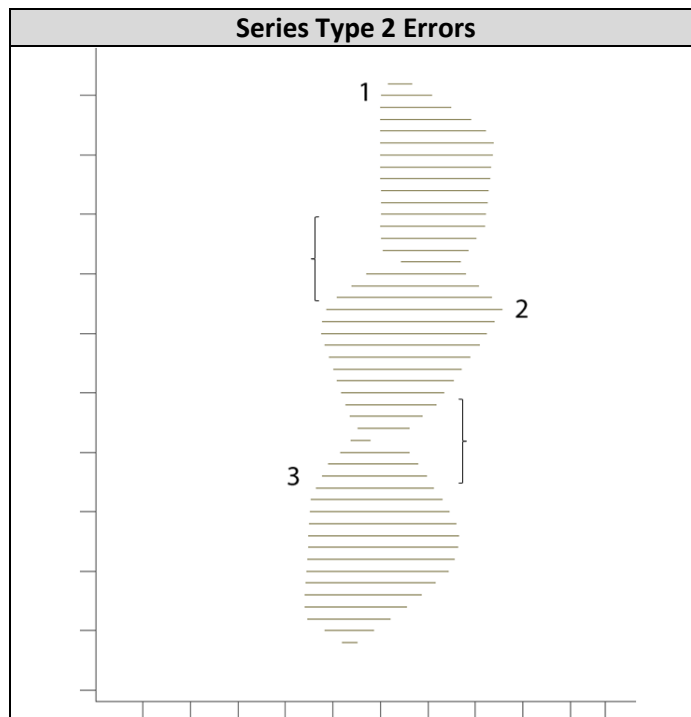


Figure 4.15: Series Type 2 error

Type 3 Error

A third type of error is shown in Figures 4.16 (a) and (b). As with Type 2 errors, Type 3 errors also affect adjacent cells. The primary difference lies with the fact that Type 3 errors correspond to incorrectly associated junctions resulting in two cell groupings. That is, with reference to Figure 4.17, association errors have occurred between slices indicated by the red markers.

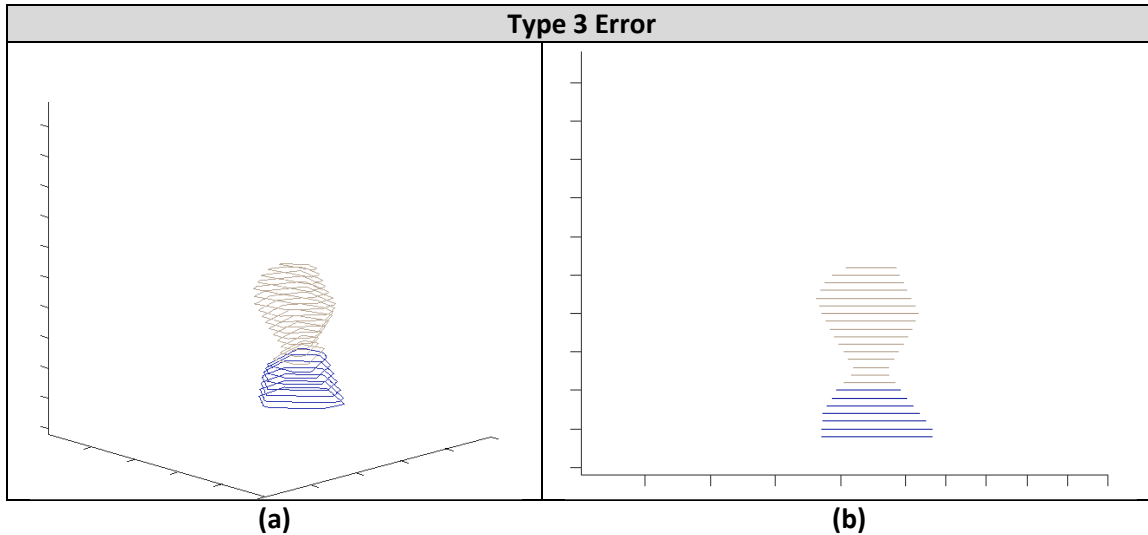


Figure 4.16: Typical Type 3 error

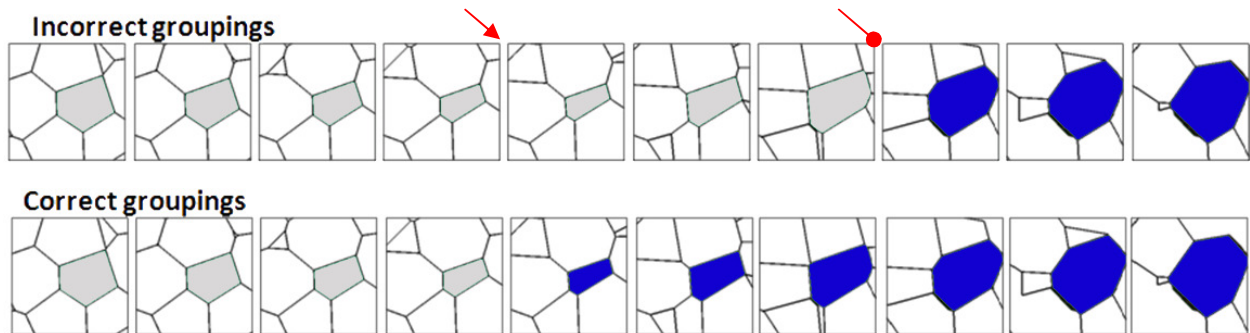


Figure 4.17: Slice-by-slice Type 3 error

In truth, Type 3 errors are a combination of Type 1 and Type 2 errors. In Figure 4.17, consider the slices indicated by the red arrow. This corresponds to a junction association error characteristic to Type 2 errors, whereas the red-rounded marker indicates a group-splitting mis-association similar to Type 1 errors. It is the combination of such errors that leads to the appearance of Type 3 errors.

4.2.3 Unassigned Contours

An additional error type that occurs is that of unassigned contours. As previously described, unassigned contours typically arise in Type 2 errors. However, some contours may remain unassigned independently, and are errors in their own right. Figure 4.18 depicts a case in which successive contours have been left unassigned. In truth, unassigned regions in slice 28 thru 32 belong to the green shaded grouping in slices 25 to 27. This is largely caused by low similarity values between adjacent contours.



Figure 4.18: Unassigned contours

4.3 Evaluation Measures

Having discussed typical error types, it is also appropriate to discuss evaluation measures. The effectiveness of the CGA will be evaluated utilizing a number of such measures. Fundamentally, the CGA establishes contour groupings through a series of individual contour associations. Thorough evaluation requires both grouping and individual association accuracy to be analyzed. For the purposes of this thesis, measures dealing with groupings will be considered to be “macro-level”, while measures dealing with individual associations will be considered “micro-level”. Given macro- and micro-level evaluations, a final “overall accuracy” will be derived to describe the effectiveness of the CGA. All measures are described in Table 4.2.

Table 4.2: Evaluation measure definition

	Definition	Describes
Grouping-Cell Match (GCM)	$1 - \left \frac{\# \text{ true cells} - \# \text{ CGA groupings}}{\# \text{ true cells}} \right $	<ul style="list-style-type: none"> ability of CGA to establish correct number of groupings
Correct Groupings (CG)	$\frac{\# \text{ correct groupings}}{\# \text{ CGA groupings}}$	<ul style="list-style-type: none"> accuracy of established groupings
Grouping Accuracy (GA)	$GCM \cdot CG$	<ul style="list-style-type: none"> overall effectiveness in establishing correct groupings
Association Accuracy (AA)	$1 - \frac{\# \text{ association errors}}{\# \text{ CGA associations}}$	<ul style="list-style-type: none"> accuracy of individual CGA contour associations
Overall Accuracy (OA)	$GA \cdot AA$	<ul style="list-style-type: none"> overall effectiveness of CGA

From Table 4.2, three measures on the macro-level will be used to evaluate the accuracy of the CGA. These are: 1) grouping-cell match (GCM), 2) correct groupings (CG) and 3) grouping accuracy (GA). Note that each measure is represented as a percentage, with grouping accuracy defined as the product of GCM and CG. Evaluation of micro-level effectiveness is obtained using “association accuracy (AA)”, while overall accuracy is the product of GA and AA.

Chapter 5: Algorithm Performance

Chapter 3 described considerations taken to develop a contour grouping algorithm (CGA). The CGA consisted of two parts: 1) a probabilistic grouping step and 2) a cost optimization step. Application of the CGA required use of multiple parameters. Initial parameter values are listed in Table 5.1.

Table 5.1: Initial Parameter Values

Adjustable Parameter	Affects	Value
Gaussian Parameters		
σ	Bending metric; Topology metric	$\frac{7\pi}{36}$
μ	Bending metric; Topology metric	0
Probabilistic Thresholds		
Triplet Threshold	Probabilistic grouping	70%
Topology Threshold	Probabilistic grouping	50%
Conditional Triplet Threshold	Probabilistic grouping	80%
Conditional Topology Threshold	Probabilistic grouping	70%
X-Y Triplet Threshold	Probabilistic grouping	85%
Cost Function Parameters		
Minimum/Maximum Pixel Overlap Cost	Cost optimization grouping	5/15
Minimum/Maximum Topology Cost	Cost optimization grouping	0/20
Minimum/Maximum Bending Cost	Cost optimization grouping	0/15
Area Difference Cost	Cost optimization grouping	10
Centroid Distance Cost	Cost optimization grouping	4

Discussion on the choice of parameter values, typical error types and evaluation measures has been presented in Chapter 4. With these in mind, two synthetic sets of data generated using custom software from the Biomechanics Lab at the University of Waterloo were analyzed. Initial analysis is followed by a brief study of parameter value changes. This discussion is not meant to be extensive, serving only to depict the robustness of the CGA.

Following this, synthetic data with varied slicing distances is analyzed. In conjunction with trials on real confocal image data, these trials assist in understanding the many factors that affect CGA performance. The two sets of confocal images were obtained from Jeff Axelrod (Stanford School of Medicine, Stanford, California) and Antonio Jacinto (Institute of Molecular Medicine, Lisbon, Portugal) respectively. The images are used with permission.

5.1 Synthetic Data

Two sets of synthetic data were generated for analysis. The first set (Figure 5.1) consists of one-hundred cells in a curved sheet, while the second set (Figure 5.4) is a fifty cell-aggregate. The curved

sheet was generated by bending an extruded two-dimensional voronoi tessellation. The cell-aggregate was similarly generated utilizing a three-dimensional voronoi. In both cases, analogous biological instances are observed in fluorescent confocal microscopy. The curved sheet may be seen as a subsection of fluorescent epithelial cells, while the cell aggregate may be a particular time instance in cell sorting.

The data sets to be examined are inherently much different from one-another. That is, the curved cell sheet represents an arrangement in which less uncertainty exists between proper contour associations. Consider that the cell aggregate consists of various layers of cells. The uncertainty caused by confocal slices is correspondingly higher since it may be difficult to determine the extent of a particular cell.

5.1.1 Curved Cell Sheet

Figure 5.1 (a) through (f) depicts six sample slices of a curved cell sheet with resolution of 491 x 555 x 71.

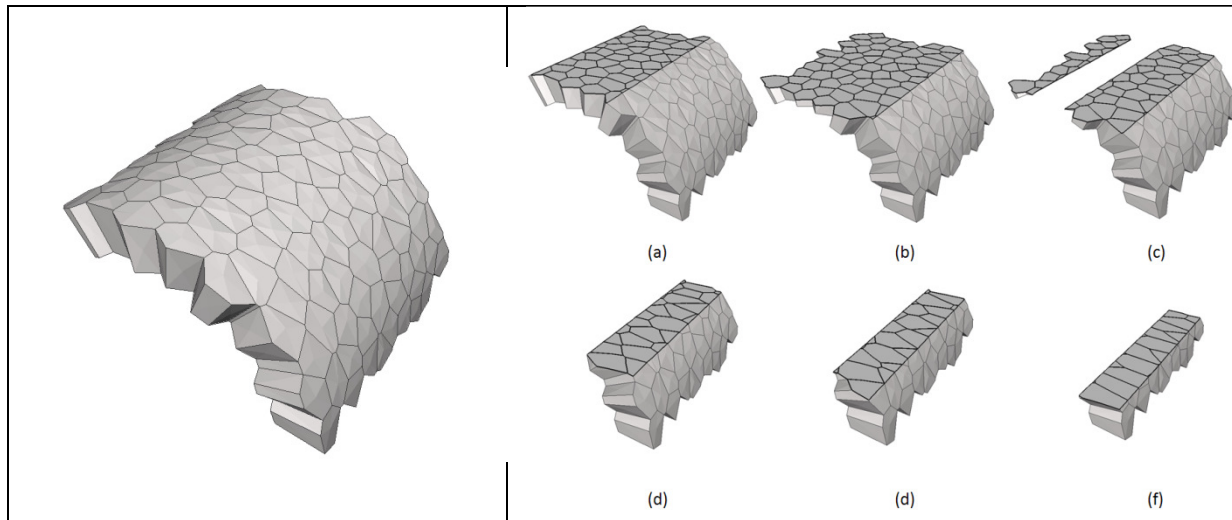


Figure 5.1: 100-cell curved sheet

The curved three-dimensional structure results in slices of various contour geometries. Defining a “cell axis” normal to a cell’s top face (Figure 5.2), cells can be classified into three classes. That is, the angle between respective cell axes and the z- axis are used to categorize cells into: 1) longitudinal, 2) transitional and 3) transverse classes (Table 5.2 and Figure 5.2). As cells belonging to different classes are sliced in the z- direction, contours of varying geometries are produced on subsequent slices.

Table 5.2: Cell orientations

Class	Angle Range
Longitudinal	0° - 30°
Transitional	30° - 60°
Transverse	60° - 90°

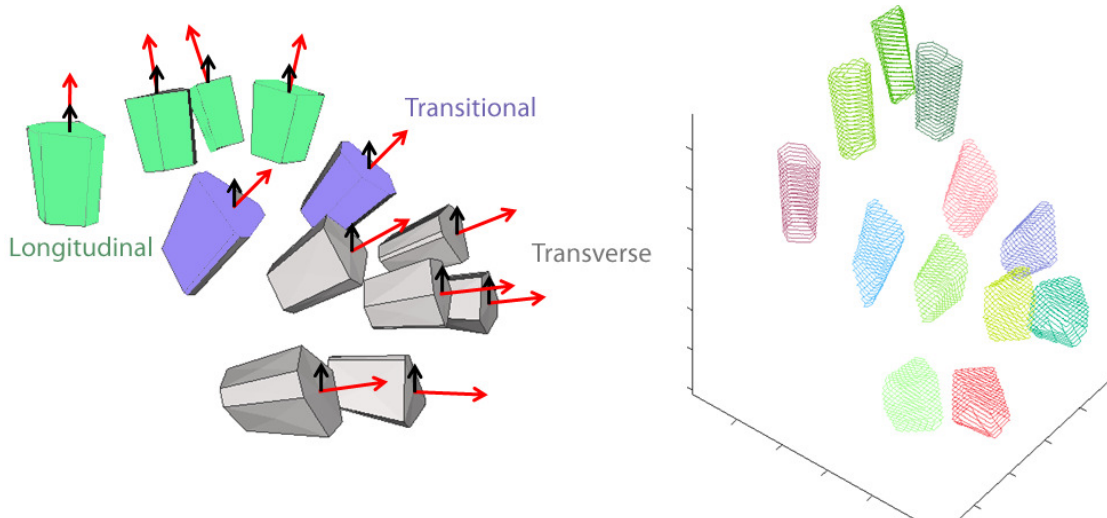


Figure 5.2: Cell orientations

The CGA is thus tasked to find associations between contours of varying geometries across slices. Ultimately, the micro-level associations are expected to result in accurate macro-level groupings. However, given the large number of cells, and varying cell orientations, it is seen that this task is not trivial.

Results

Figure 5.3 (a) depicts the CGA output for this set of synthetic data. The diagram portrays cell groupings made after all iterations of the CGA. Visually, the algorithm has performed well. However, close inspection reveals a total of 101 groupings have been established. Since the true number of cells is 100, there exists an extra grouping. Further investigation reveals that the extra grouping is caused by a single Type 1 Error in Figure 5.3 (b). Complete algorithm statistics are provided in Table 5.3.

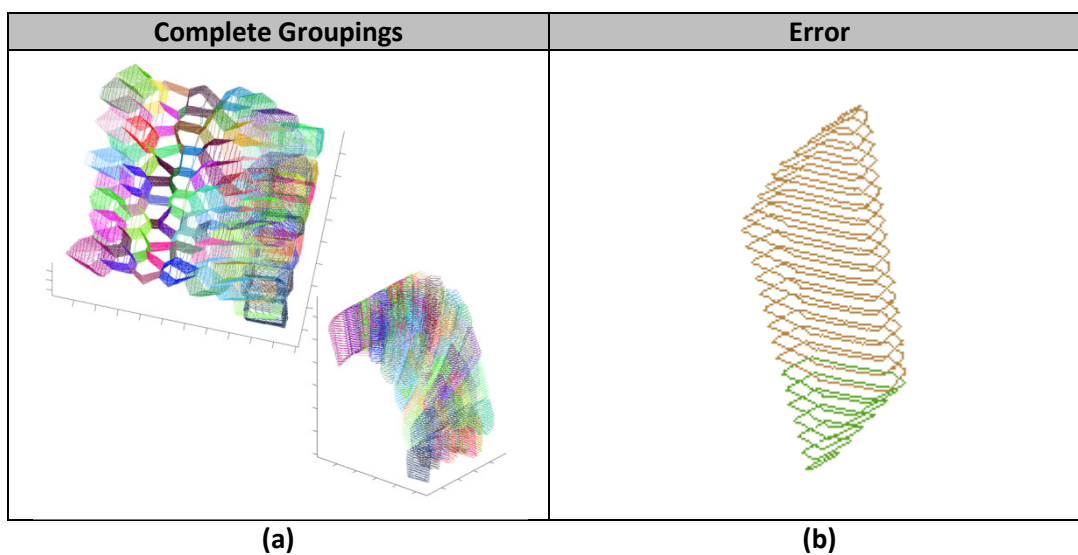


Figure 5.3: Curved cell sheet results

Table 5.3: Curved cell sheet algorithm statistics

Algorithm Statistics			
<i>Actual Number of Groupings:</i>	100	<i>Number of 2D Contours:</i>	2144
<i>Number of Groupings Found:</i>	101	<i>Number of Probabilistic Iterations:</i>	2
<i>Number of Correct Groupings:</i>	99	<i>Number of Type 1 Optimization Iterations:</i>	13
<i>Number of Associations:</i>	2043	<i>Number of Type 2 Optimization Iterations:</i>	2
<i>Correct Associations:</i>	2042	<i>Run Time:</i>	51.95s

From a total of 2144 two-dimensional contours, a total of 99 correct three-dimensional groupings were established. Furthermore, manual analysis found that a single Type 1 error existed, contributing to the lone extra grouping. Of the 2043 total contour associations, only 1 is in error. In total, the CGA conducted 2 probabilistic iterations, coupled with 15 optimization iterations. The total run-time was 51.95s.

Evaluation

From the statistics provided in Table 5.3, evaluation measures were calculated (Table 5.4). For this data set, the CGA has performed well, with an overall accuracy of 97%. The CGA attained 97% grouping accuracy, while 98% of established groupings were correct. This result is encouraging, as it indicates that the CGA is capable of establishing a large number of correct groupings. Furthermore, despite the Type 1 error, the CGA shows promise in associating curved cell sheets. Specifically, since curved sheets mimic the geometry of fluorescent epithelial cells, the CGA may be effective in analyzing real image data of such types.

Table 5.4: Curved cell sheet evaluations

Grouping-Cell Match	Correct Groupings	Grouping Accuracy	Association Accuracy	Overall Accuracy
99%	98%	97%	~100%	97%

5.1.2 Cell Aggregate

Figure 5.4 (a) through (f) depicts six sample slices of a 50 cell aggregate with resolution of 500 x 500 x 56. This aggregate is geometrically comparable to those obtained in cell sorting experiments and indicates that successive slices of the same cell produce contours of varying geometries. Unlike the curved cell sheet, the aggregate contains multiple layers of cells. Consequently, progressive slices depict cells disappearing or appearing based on their extents.

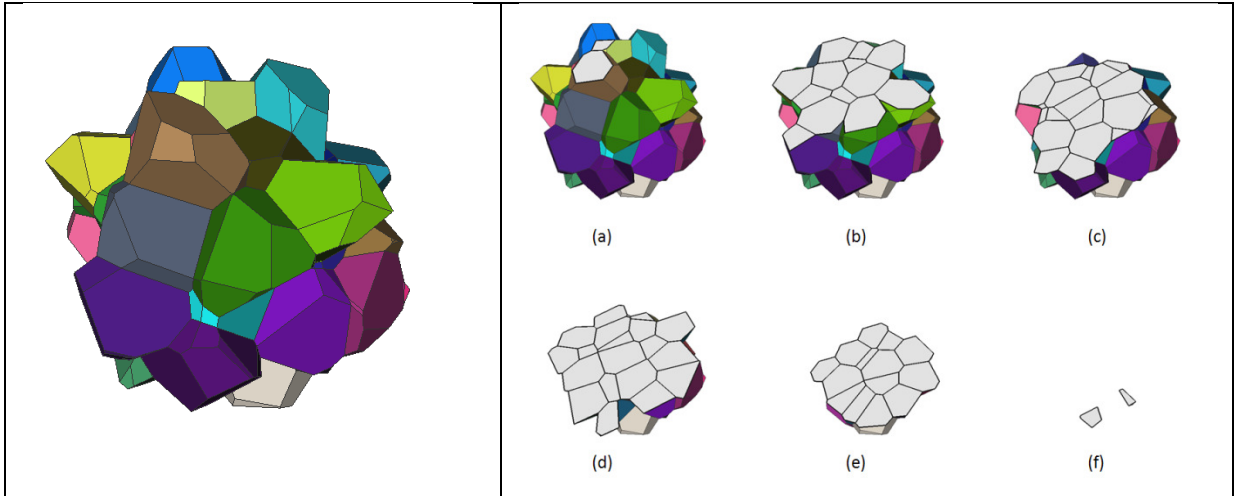


Figure 5.4: 50-cell aggregate

Results

Figure 5.5 shows the result of the CGA. cursory examination of colorized contours indicates accurate groupings have been obtained. Note however that only 46 groupings have been established. Analysis reveals that a number of errors have occurred. These are summarized in Table 5.5, with a total of 16 errors (including unassigned contours). Algorithm statistics are provided in Table 5.6.

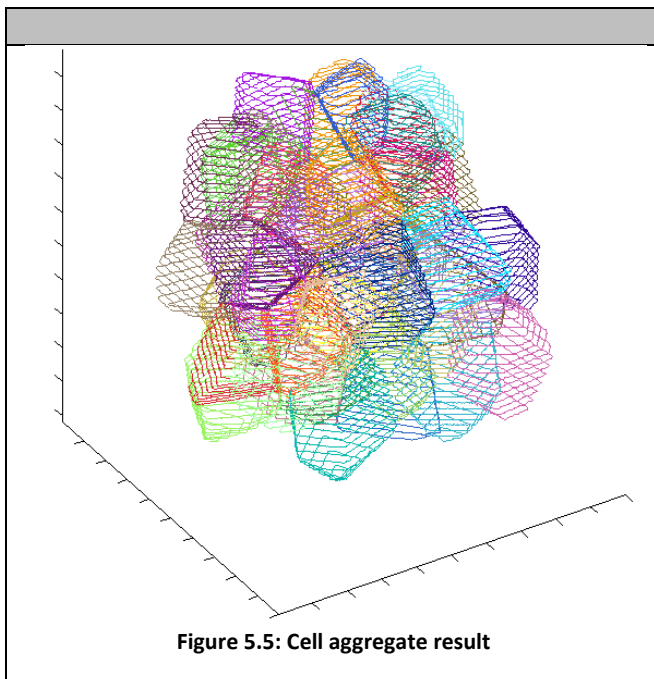


Figure 5.5: Cell aggregate result

Table 5.5: Cell aggregate errors

Number of Type 1 Errors	1
Number of Type 2 Errors	4
Number of Type 3 Errors	1
Number of Unassigned Regions	9
Total Number of Errors	16

Table 5.6: Cell aggregate algorithm statistics

Algorithm Statistics			
<i>Actual Number of Groupings:</i>	50	<i>Number of 2D Contours:</i>	730
<i>Number of Groupings Found:</i>	46	<i>Number of Probabilistic Iterations:</i>	2
<i>Number of Correct Groupings:</i>	39	<i>Number of Type 1 Optimization Iterations:</i>	9
<i>Number of Associations:</i>	675	<i>Number of Type 2 Optimization Iterations:</i>	2
<i>Correct Associations:</i>	668	<i>Run Time:</i>	25.42s

From 730 two-dimensional contours, a total of 39 correct three-dimensional groupings were established. Of the 675 total contour associations, 17 were incorrect. In total, the CGA conducted 2 probabilistic iterations, coupled with 11 optimization iterations. The total run-time was 25.42s.

Evaluation

Evaluation measures for the aggregate are as shown in Table 5.7. These measures indicate the CGA is less effective in associating and grouping a cell aggregate. Nonetheless, overall accuracy of 77.2% is acceptable. Note that individual associations have been made surprisingly well. That is, 99% of contour associations were made correctly. However, 1% of incorrect associations led to a severe degradation in the ability of the CGA to establish accurate groupings (78% grouping accuracy).

Table 5.7: Cell aggregate evaluations

Grouping-Cell Match	Correct Groupings	Grouping Accuracy	Association Accuracy	Overall Accuracy
92%	84.8%	78%	99%	77.2%

5.1.3 Discussion

The results presented above suggest the CGA is highly effective in establishing correct cell groupings from confocal image stacks. The simulations presented prove that the CGA is capable of dealing with various data sets (i.e. sheet or aggregate). However, the CGA is markedly stronger in establishing groupings from cell sheets (97% accuracy) as opposed to cell aggregates (77.2% accuracy). Although only a single simulation for each data set has been presented, it is confidently stated that the architecture of the CGA is robust.

Interestingly, micro-level measures indicate the CGA is highly effective (> 99% accurate) in establishing individual contour associations. However, the small number of micro-level errors contributes to a significantly larger number of macro-level errors. That is, a disproportionate number of incorrect cell groupings occur despite a small number of incorrectly associated contours.

5.2 Parameter Changes

With the observations made in Section 5.1.3 in mind, it is natural to investigate whether changes in parameter values may alleviate errors produced by the CGA. Given parameters in Table 5.1, do modifications to particular parameters significantly modify CGA performance? This section aims to address this concern. Performance is evaluated for a set of 10 different parameter settings. Note that modification of parameters causes specific changes in CGA behavior. These affects are described in Table 5.8. Specific parameter sets are given in Table 5.9.

Table 5.8: Parameter Affects

Parameter	Affects
σ	Bending similarity
Trip. Threshold	Initial strong triplet acceptance
Top. Threshold	Strong triplet topology check
Cond. Trip. Threshold	Z-extension
Cond. Top. Threshold	Z-extension topology check
X-Y Trip. Threshold	X-Y extension
Min./Max. Pixel Overlap Cost	Grouping cost evaluation
Min./Max. Topology Cost	Grouping cost evaluation
Min./Max. Bending Cost	Grouping cost evaluation
Area Difference Cost	Grouping cost evaluation
Centroid Distance Cost	Grouping cost evaluation

Table 5.9: Parameter change cases

Parameter	Case A	Case B	Case C	Case D	Case E	Case F	Case G	Case H	Case I	Case J	Case K
	Original	Topology Threshold Change			X-Y Triplet Threshold Change			Combination Change	Cost Function Change		
σ	0.611	---	---	---	---	---	---	---	---	---	---
μ	0	---	---	---	---	---	---	---	---	---	---
Trip. Threshold	0.7	---	---	---	---	---	---	---	0.9	---	---
Top. Threshold	0.5	0.7	0.9	0.3	---	---	---	0.9	0.9	---	---
Cond. Trip. Threshold	0.8	---	---	---	---	---	---	---	0.9	---	---
Cond. Top. Threshold	0.7	---	---	---	---	---	---	0.9	0.9	---	---
X-Y Trip. Threshold	0.85	---	---	---	0.7	0.95	0.50	0.9	---	---	---
Min./Max. Pixel Overlap Cost	5/15	---	---	---	---	---	---	---	---	5/15	0/10
Min./Max. Topology Cost	0/20	---	---	---	---	---	---	---	---	5/15	20/50
Min./Max. Bending Cost	0/15	---	---	---	---	---	---	---	---	5/15	20/50
Area Difference Cost	10	---	---	---	---	---	---	---	---	---	---
Centroid Distance Cost	4	---	---	---	---	---	---	---	---	---	---

Parameter values for initial simulations presented in Section 5.4 are given as Case A. The remaining 10 parameter sets fall within 4 distinct categories: 1) topology threshold change, 2) X-Y threshold change, 3) combination change and 4) cost function change. As stated earlier, parameter changes are not meant to constitute an extensive parametric study. Rather, it serves as an introduction to the variable behavior of the CGA.

5.2.1 Curved Cell Sheet

Results

Table 5.10 summarizes the results of all simulations with parameter value changes. For the curved cell sheet, it is seen that the CGA is fairly robust. That is, referring to the accuracy of the CGA, each simulation resulted in more than 90 correct groupings. Additionally, it is seen that the number of errors in each case is relatively small. However, note that efficiency statistics of the CGA varies. That is, despite consistency in accuracy, the CGA arrives at contour groups through a different path. The number of iterations for each portion of the CGA differs from case to case.

Table 5.10: Parameter change results -- cell sheet

<i>Algorithm Statistics - Curved</i>	A	B	C	D	E	F	G	H	I	J	K
	Original	Topology Threshold Change			X-Y Triplet Threshold Change			Combination Change		Cost Function Change	
Accuracy											
Actual Number of Groupings:	100	100	100	100	100	100	100	100	100	100	100
# 2D Contours:	2144	2144	2144	2144	2144	2144	2144	2144	2144	2144	2144
# Associations:	2043	2044	2037	2044	2044	2044	2043	2042	2036	2036	2036
# Association Errors:	1	0	7	0	0	0	1	2	8	8	8
# Groupings Found:	101	100	107	100	100	100	101	102	108	108	108
# Correct Groupings:	99	100	93	100	100	100	99	98	92	92	92
Efficiency											
Probabilistic Iterations:	2	2	2	2	2	2	2	2	1	2	2
Type 1 Optimization Iterations:	13	11	14	12	9	8	9	14	12	7	12
Type 2 Optimization Iterations:	2	2	2	2	2	3	2	2	2	2	2
Run Time:	51.95	49.43	58.02	43.1	37.65	38.69	38.91	75.12	288.9	33.71	49.7
Total Triplets:	33824	33824	33824	33824	33824	33824	33824	33824	33824	33824	33824
Probability Accepted Triplets:	1223	1223	1223	1223	1223	1223	1223	1223	671	1223	1223
Topology Accepted Triplets:	1214	1192	928	1216	1214	1214	1214	928	610	1214	1214
Errors											
Type 1 Errors:	1	0	7	0	0	0	1	2	8	8	8
Type 2 Errors:	0	0	0	0	0	0	0	0	0	0	0
Type 3 Errors:	0	0	0	0	0	0	0	0	0	0	0
Unassigned Contours:	0	0	0	0	0	0	0	0	0	0	0

Additionally, it is seen that modification of topology thresholds changes the number of topology accepted triplets. Changes in X-Y threshold do not produce significant changes in statistics, while modifications involving the triplet threshold cause a change in the number of probability accepted triplets. For example, in Case I, the triplet threshold is increased from 0.7 to 0.9. Accordingly, the number of probability accepted triplets decreased from 1223 to 671. As a result of this, the run time of the CGA increased from 51.95s to 288.9s. This underlines that acceptance of initial strong triplets is a large factor in determining efficiency of the CGA.

Evaluation

In terms of CGA effectiveness, evaluation measures presented in Table 5.11 shows that overall accuracy for the curved cell sheet is high. For 7 of the 11 cases, the CGA attains greater than 90% overall accuracy. In those cases where overall accuracy is below 90%, association accuracy is still above 99%. Thus, as with Section 5.1, a small number of association errors lead to a much larger number of grouping errors.

Table 5.11: Parameter change evaluations -- cell sheet

<i>Evaluation - Curved</i>	A	B	C	D	E	F	G	H	I	J	K
Groupings-Cell Match:	99%	100%	93%	100%	100%	100%	99%	98%	92%	92%	92%
Correct Groupings:	98%	100%	87%	100%	100%	100%	98%	96.1%	85.2%	85.2%	85.2%
Grouping Accuracy:	97%	100%	81%	100%	100%	100%	97%	94.2%	78.4%	78.4%	78.4%
Association Accuracy:	100%	100%	99.7%	100%	100%	100%	100%	99.9%	99.6%	99.6%	99.6%
Overall Accuracy:	97%	100%	80.6%	100%	100%	100%	97%	94.1%	78.1%	78.1%	78.1%

Additionally, the three least accurate cases are Cases I, J and K. Case I represents an instance in which probability thresholds are set unreasonably high (i.e.: above 90%) while Cases J and K are cost function modifications. For Case I, the loss in accuracy may be attributed to unreasonable expectations, while for Cases J and K, loss of accuracy suggests that modification of the cost function is ill-advised. In this sense, it is seen that the CGA, despite being robust, does depend upon the selection of suitable parameter values and cost functions. However, from the 11 trials presented, the CGA remains accurate for a variety of parameter modifications.

Interestingly, considering Cases C and H, it is seen that various parameters are interrelated. That is, in Case C, modification of topology threshold to 0.9 caused a decrease in overall accuracy from 97% (Case A) to 80.6% (Case C). However, in Case H, where modifications to conditional topology and X-Y triplet thresholds are included, overall accuracy was increased to 94.1%. This suggests that modifications to CGA parameters may need to be considered in groups to optimize performance.

5.2.2 Cell Aggregate

Results

The results of the CGA on the cell aggregate are presented in Table 5.12. With the exception of Case I, the number of correct groupings for each case is quite similar to the original Case A. Cases E, G, H and J result in a larger number of correct groupings, suggesting that parameter values for those cases are better suited to dealing with the cell aggregate. Similar to 5.2.1, differing sets of parameters results in variable CGA efficiency, with each case requiring a different number of CGA iterations. Also note the marked increase in run time for Case I. In this instance, only 2 topology accepted triplets occurred, suggesting the majority of contour associations and groupings were made in the cost optimization phase of the CGA.

Table 5.12: Parameter change results -- cell aggregate

<i>Algorithm Statistics - Aggregate</i>	A	B	C	D	E	F	G	H	I	J	K
	Original	Topology Threshold Change			X-Y Triplet Threshold Change			Combination Change	Cost Function Change		
Accuracy											
Actual Number of Groupings:	50	50	50	50	50	50	50	50	50	50	50
# 2D Contours:	730	730	730	730	730	730	730	730	730	730	730
# Associations:	675	676	673	674	674	674	671	671	649	674	661
# Association Errors:	7	7	6	6	4	5	7	7	25	4	15
# Groupings Found:	46	44	45	47	47	47	51	49	75	47	60
# Correct Groupings:	39	37	38	39	42	39	43	40	24	42	32
Efficiency											
Probabilistic Iterations:	2	2	2	2	2	2	2	2	1	2	2
Type 1 Optimization Iterations:	9	11	8	14	13	13	7	21	4	10	12
Type 2 Optimization Iterations:	2	3	2	2	2	2	2	2	4	2	2
Run Time:	25.42	37.38	79.56	34.95	33.76	36.05	25.12	53.9	96.49	38.41	38.59
Total Triplets:	11488	11488	11488	11488	11488	11488	11488	11488	11488	11488	11488
Probability Accepted Triplets:	230	230	230	230	230	230	230	230	14	230	230
Topology Accepted Triplets:	221	194	179	226	221	221	221	179	2	221	221
Errors											
Type 1 Errors:	1	0	0	1	0	0	2	2	24	0	12
Type 2 Errors:	4	7	6	5	4	5	3	5	1	4	3
Type 3 Errors:	1	0	0	0	0	0	1	0	0	0	0
Extra Groupings:	1	1	1	1	1	2	1	2	2	1	1
Unassigned Contours:	9	10	12	9	9	9	8	10	6	9	9

Evaluation

Evaluation measures presented in Table 5.13 indicate that, in 9 instances, the CGA performs above 70% accurately. Since the cell aggregate has been identified to be more complex than the cell sheet,

it is not surprising that the CGA is less accurate in this case. However, note that association accuracy for all cases remains extremely high (i.e.: > 96%). It is thus reiterated that relatively few association errors lead to a large number of incorrect groupings.

Table 5.13: Parameter change evaluations -- cell aggregate

<i>Evaluation - Aggregate</i>	A	B	C	D	E	F	G	H	I	J	K
Groupings-Cell Match:	92%	88%	90%	94%	94%	94%	98%	98%	50%	94%	80%
Correct Groupings	84.8%	84.1%	84.4%	83%	89.4%	83%	84.3%	81.6%	32%	89.4%	53.3%
Grouping Accuracy:	78%	74%	76%	78%	84%	78%	82.6%	80%	16%	84%	42.7%
Association Accuracy:	99%	99%	99.1%	99.1%	99.4%	99.3%	99%	99%	96.1%	99.4%	97.7%
Overall Accuracy:	77.2%	73.2%	75.3%	77.3%	83.5%	77.4%	81.8%	79.2%	15.4%	83.5%	41.7%

The robustness of the CGA is once again seen, as differing parameter sets maintain a suitable level of accuracy for the majority of cases. The low overall accuracy of Case I (15.4%) suggests that reliance on the cost portion of the CGA may lead to low grouping accuracy (16%). However, given that association accuracy for this case was 96.1%, it is difficult to classify this CGA trial as a failure. Rather, Case I shows that the CGA is very strong in producing correct associations (micro-level) but is much weaker in establishing correct contour groupings (macro-level). Comparison of Case C with Case H once again shows that parameter changes may become more effective in groups (~4% increase in overall accuracy).

5.2.3 Observations

The trials presented in Sections 5.2.1 and 5.2.2 show that the CGA is robust. The CGA consistently establishes accurate groupings despite variation of parameter values. For the curved cell sheet, overall accuracy remained above 75% for all trials. For the cell aggregate, 9 of 11 trials attained higher than 70% overall accuracy. As such, it can be said that the CGA is effective in finding contour groupings from confocal image stacks.

Referring to Table 5.9, it is seen that Case I represents an extremely strict set of probabilistic thresholds. The results for this parameter set shows that the CGA typically performs more effectively under moderate constraints. That is, since attaining proper groupings from confocal images is, by nature, a fuzzy task, over-constraining of probabilistic thresholds is undesirable. For the curved cell sheet, accuracy remains high at 78.1%, but is due to the relative certainty with which certain contour groups in the sheet exhibit. For the cell aggregate, the level of uncertainty is higher (due to overlapping layers of cells), and thus, over-constraining the problem leads to severe degradation of accuracy (15.4%).

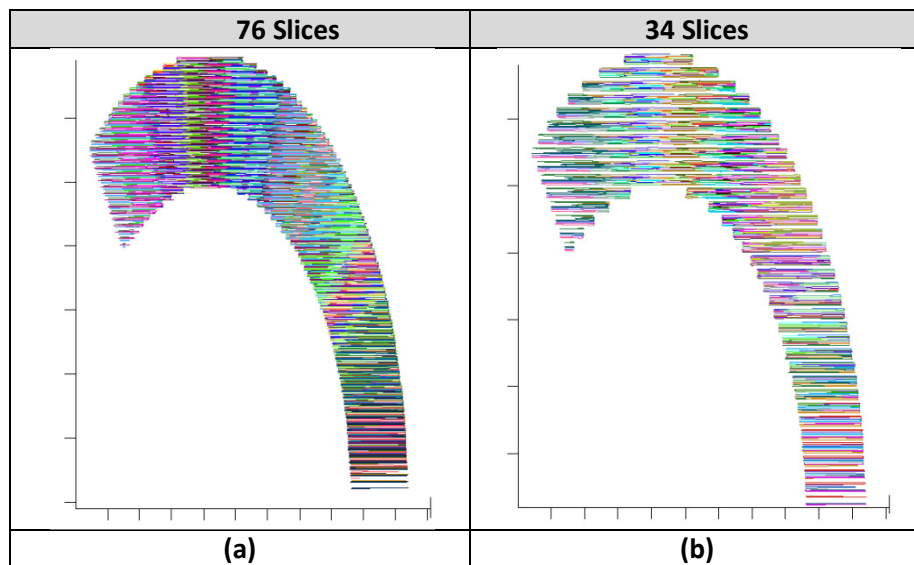
On the micro-level, individual association accuracy for all trials was in excess of 96%. All but one trial (Case I – cell aggregate), was above 99% accuracy. Thus, the CGA is extremely strong on the micro-level. However, correspondingly lower grouping accuracies indicate that macro-level

effectiveness is highly sensitive to micro-level errors. Despite grouping errors, the CGA framework is fundamentally strong.

5.3 Slicing Distance Changes

Results presented in Sections 5.1 and 5.2 indicate the CGA is extremely effective in associating two-dimensional contours across slices. Furthermore, the CGA is fairly strong in establishing correct three-dimensional groupings. The CGA framework is fundamentally strong and is highly customizable. However, issues such as slice distance, image magnification, image contrast, etc. have yet to be addressed. For example, given a set of three-dimensional data, does its z- resolution have a large effect on CGA performance? This section explores this consideration. Specifically, the curved cell sheet will be analyzed by varying the number of synthetic slices taken. The purpose of these trials is to demonstrate that CGA performance may be affected by imaging protocols (i.e. slicing, resolution, magnification, etc.).

Z- resolution defines the “fineness” to which image sets are taken. That is, as shown in Figure 5.6, number of slices directly influences the amount of information available to the CGA. Conceptually, a larger number of slices results in “better” associations, albeit with added computational overhead.



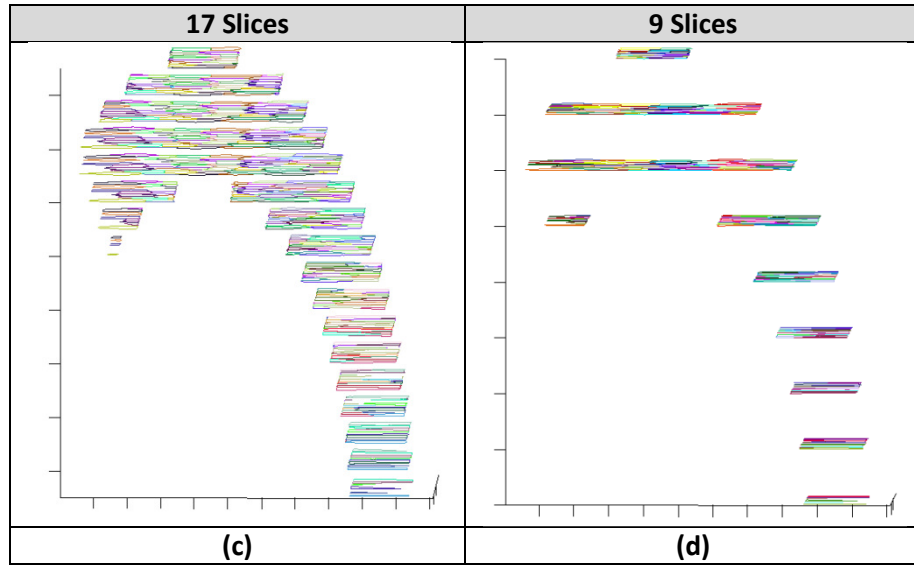


Figure 5.6: Slicing differences

For comparison purposes, the initial trial for the curved cell sheet is provided as Case A in Table 5.15. The initial trial consisted of 76 evenly spaced slices. Subsequent trials for 34, 17 and 9 evenly spaced slices are presented as Cases B, C and D respectively. Parameter values are as given in Table 5.14.

Table 5.14: Slicing parameters

Adjustable Parameter	Affects	Value
σ	Bending metric; Topology metric	$\frac{7\pi}{36}$
μ	Bending metric; Topology metric	0
Triplet Threshold	Probabilistic grouping	70%
Topology Threshold	Probabilistic grouping	50%
Conditional Triplet Threshold	Probabilistic grouping	80%
Conditional Topology Threshold	Probabilistic grouping	70%
X-Y Triplet Threshold	Probabilistic grouping	85%
Minimum/Maximum Pixel Overlap Cost	Cost optimization grouping	5/15
Minimum/Maximum Topology Cost	Cost optimization grouping	0/20
Minimum/Maximum Bending Cost	Cost optimization grouping	0/15
Area Difference Cost	Cost optimization grouping	10
Centroid Distance Cost	Cost optimization grouping	4

5.3.1 Results

Results of this trial are presented in Table 5.15.

Table 5.15: Slicing results

<i>Algorithm Statistics - Slicing</i>	A	B	C	D
	76 Slices	34 Slices	17 Slices	9 Slices
Accuracy				
Actual Number of Groupings:	100	100	100	99
# 2D Contours:	2144	1073	536	264
# Associations:	2043	967	408	162
# Association Errors:	1	3	25	9
# Groupings Found:	101	97	125	99
# Correct Groupings:	99	94	75	86
Efficiency				
Probabilistic Iterations:	2	2	2	1
Type 1 Optimization Iterations:	13	6	2	1
Type 2 Optimization Iterations:	2	2	2	2
Run Time:	51.95	54.72	52.52	32.72
Probability Accepted Triplets:	1223	366	89	8
Topology Accepted Triplets:	1214	354	82	7
Total Triplets:	33824	16656	8112	3760
Errors				
Type 1 Errors:	1	0	25	2
Type 2 Errors:	0	3	0	7
Type 3 Errors:	0	0	0	0
Extra Groupings:	0	0	0	1
Incomplete Groupings:	0	0	0	1
Unassigned Contours:	0	9	3	3

As a consequence of slice reduction, the number of 2D contours available for analysis is decreased. This is not surprising, but contributes to a corresponding decrease in the number of triplets for each case. However, note that the run time of the CGA does not necessarily decrease with number of 2D contours. This may be attributed to an increased difficulty for the CGA to find strong triplets, thereby rendering it more difficult to complete contour associations. To this end, notice that the number of accepted triplets (i.e.: probability/topology) dramatically decreases with number of slices.

Nonetheless, the results indicate that the CGA is capable of finding the majority of cell groupings. That is, a minimum of 75 correct groupings were found. Curiously, as the number of slices is reduced, Type 2 errors begin to appear. In previous trials (Table 5.10), CGA analysis of the curved cell sheet had exclusively resulted in Type 1 errors. The appearance of Type 2 errors suggests that reduced slicing distance leads to CGA ambiguities not previously encountered for the cell sheet.

Additionally, visual analysis of Case D revealed a hybrid Type 1 and Type 2 error (Figure 5.7). The correct groupings are presented in Figure 5.8. The hybrid error was not previously encountered, and suggests that unexpected CGA behaviours may result from reduced number of slices. Indeed, Figures 5.7 and 5.8 depict contour groupings consisting of 3 contours. In truth, since the CGA heavily relies upon the identification of strong “triplets”, it is better suited to analyze data in which contour groupings consist of more than 3 contours.

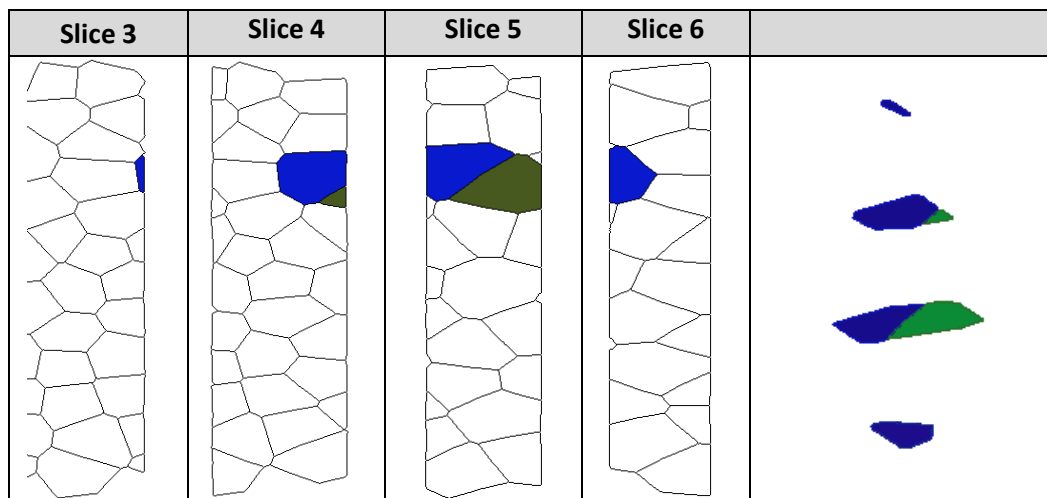


Figure 5.7: Hybrid error

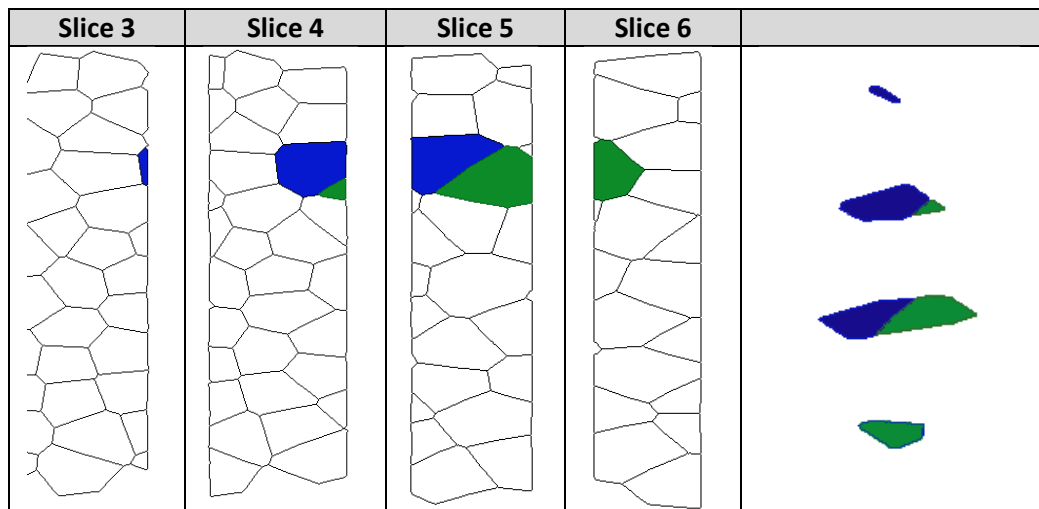


Figure 5.8: Hybrid error resolution

5.3.2 Evaluation

In general, the accuracy of the CGA in dealing with changes in the number of slices seems erratic (Table 5.16).

Table 5.16: Slicing evaluations

<i>Evaluation - Slicing</i>	A	B	C	D
Groupings-Cell Match:	99%	97%	75%	100%
Correct Groupings:	98%	96.9%	60%	86.9%
Grouping Accuracy:	97%	94%	45%	86.9%
Association Accuracy:	100%	99.7%	93.9%	94.4%
Overall Accuracy:	97%	93.7%	42.2%	82%
% Triplets Accepted:	35.9%	21.3%	1%	0.2%

In Cases A, B and C, in which the number of slices were 76, 34 and 17 respectively, overall accuracy decreased. However, a further reduction to 9 slices resulted in an increase in overall accuracy. Reduction to less than 9 slices resulted in overall CGA failure. However, comparing these results to those in Table 5.13 reveals that, in general, CGA analysis of the data set with 76 slices resulted in highly accurate groupings. This may indicate that a reduction in slice number decreases the ability of the CGA to arrive at accurate groupings. However, this is hardly convincing, since the 82% overall accuracy of Case D does not support this conclusion.

On the other hand, if the CGA is judged based on association accuracy, results from Table 5.16 suggest that decreasing the number of slices reduces effectiveness. That is, for trials presented in Table 5.13, association accuracy routinely surpassed 99% whereas for Cases A, B, C and D in Table 16, association accuracy varied from 100%, 99.7%, 93.9% and 94.4% respectively. The CGA produces more errors when dealing with less densely sliced samples.

Given that reduced slice number contributes to lower association accuracy, it is implied that grouping accuracy is that which contributes to erratic overall accuracy. The question arises as to how decreased micro-level associations can still lead to accurate macro-level groupings. As discussed in Section 5.2, there seems to be a disproportionate relationship between association accuracy and grouping accuracy. The results presented in this section suggest that, in addition to this, another factor (i.e. slicing distance), also affects this relationship. As such, it is seen that fluctuations in image protocol/quality may lead to unexpected CGA behaviours.

5.4 Real Image Data

Two sets of real image data were chosen for verification. The first set corresponds to confocal images of a developing fly-wing, with the slice direction corresponding to the physical z- axis. The second set of images depicts a region adjacent to a laser-induced epithelial wound on a drosophila embryo. Prior to input to the CGA, all images were analyzed using custom edge detection software from the Biomechanics Lab at the University of Waterloo. Note that edge tracings obtained from such software is imperfect, resulting in grouping artifacts that will be discussed. Figure 5.9 depicts the process of obtaining cell groupings.

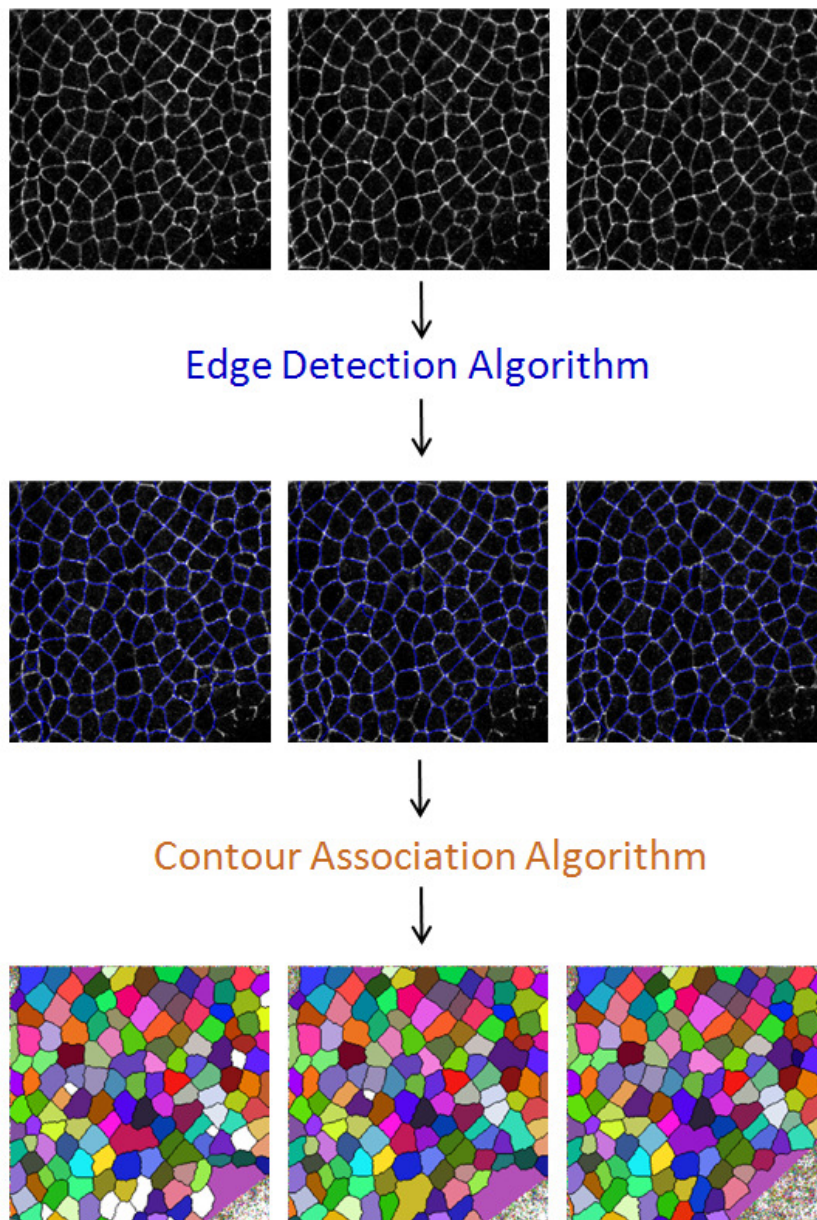


Figure 5.9: Real data analysis steps

The raw image is analyzed using edge detection software. The resulting region outlines are then input into the CGA, producing a colourized cell map. This process utilizes two pieces of custom built software (i.e. edge detection and contour grouping algorithms), and is therefore susceptible to errors in both. However, results in the following sections indicate the CGA is capable of providing accurate groupings despite imperfect edge detection.

5.4.1 Fly-wing Z- Series

The images provided in Table 5.17 are magnified in Figures 5.10 through 5.13. Raw images have an image resolution of 314 x 158 x 10 and physical resolution of 155nm x 155nm x 300.1nm per voxel. The fluorescent cell membranes appear solid but are not completely detected by edge detection algorithms. Imperfect edge tracings are input directly into the CGA.

Table 5.17: Fly-wing results overview

Raw Image					Computer Traced Outlines					Colourized Cell Groupings				

Results

Discussion of image qualities precedes discussion of CGA results. Particularly, two regions in Figure 5.10 have been highlighted. Progressing through slices (a through j), the area highlighted by A is difficult to interpret. Expression of fluorescent protein is strong, but does not represent the membrane of a particular cell. Analysis of the corresponding area in Figure 5.11 shows that edge tracings have failed. In Figure 5.12 (a) and (b), the unusual fluorescence has little impact. However, from Figure 5.12 (c) onwards, the tracings become unreliable. Referring to highlight A in Figure 5.11, the failed edge tracings contribute to incorrect groupings starting in (f).

Similarly, highlight B indicates an area in which curious membrane fluorescence is observed. Here, a strongly defined region slowly divides into two distinct ones. This process occurs in Figure 5.10 (a) to (e), with the region splitting in (f). This occurrence may correspond to layered cells similar to those in the aggregate of Section 5.2.2. Interestingly, the boundary corresponding to the appearing cell is seen initially as a “blur” at the bottom-tip of the original contour. This “blur” slowly propagates upwards in the y- direction and gains definition, resulting in a solid edge in Figure 5.10 (f). Unlike highlight A, the region indicated by B is correctly grouped by the CGA.

In addition, various region outlines are highlighted with dashed red lines in Figure 5.12. These correspond to the “appearance” of a region on a particular slice, as detected by the edge detection algorithm. These “appearing” regions underline the need for improved edge detection, as they visibly exist on previous slices but are not detected. Regardless, once found, they are correctly grouped in subsequent slices (Figure 5.13).

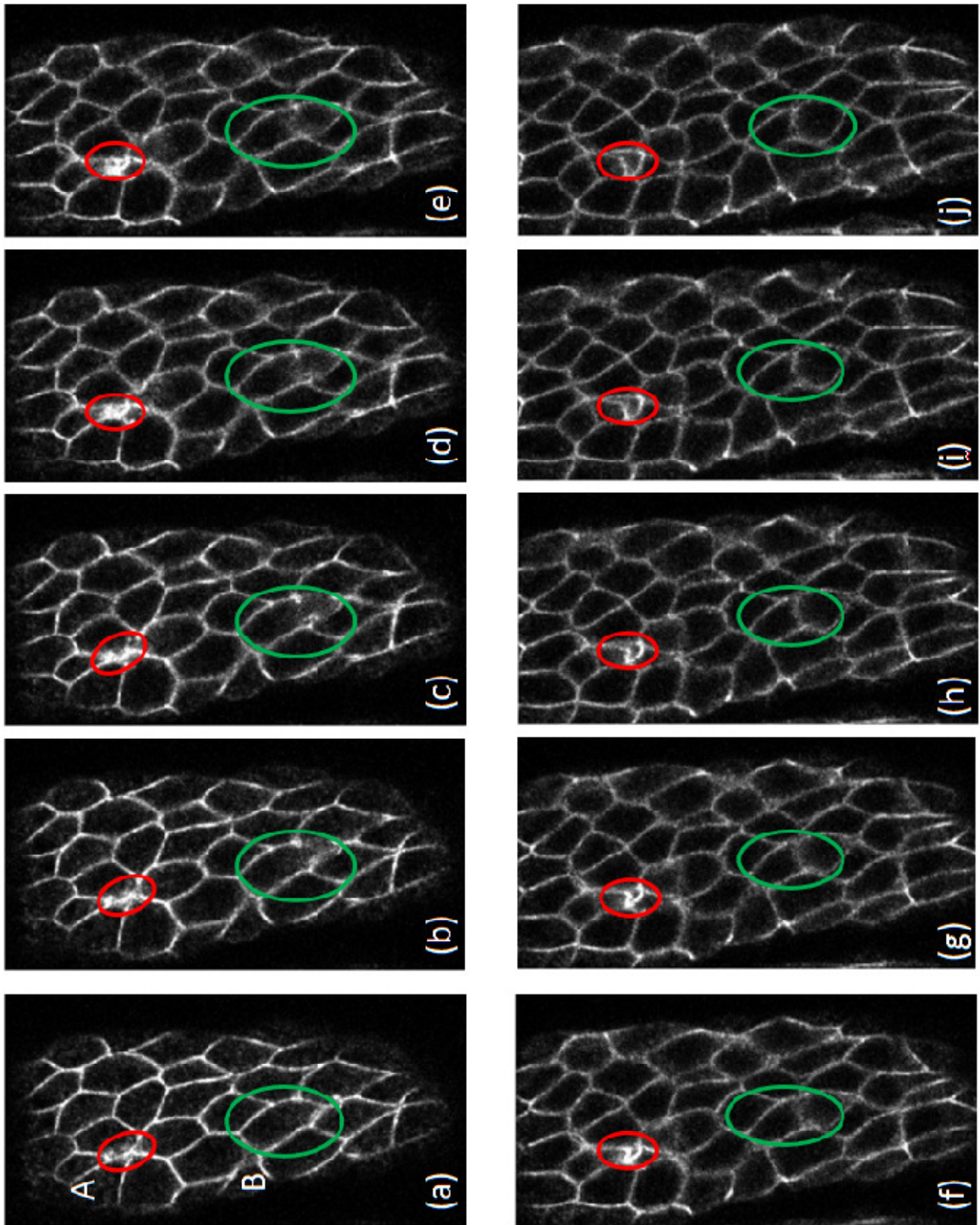


Figure 5.10: Image artifacts

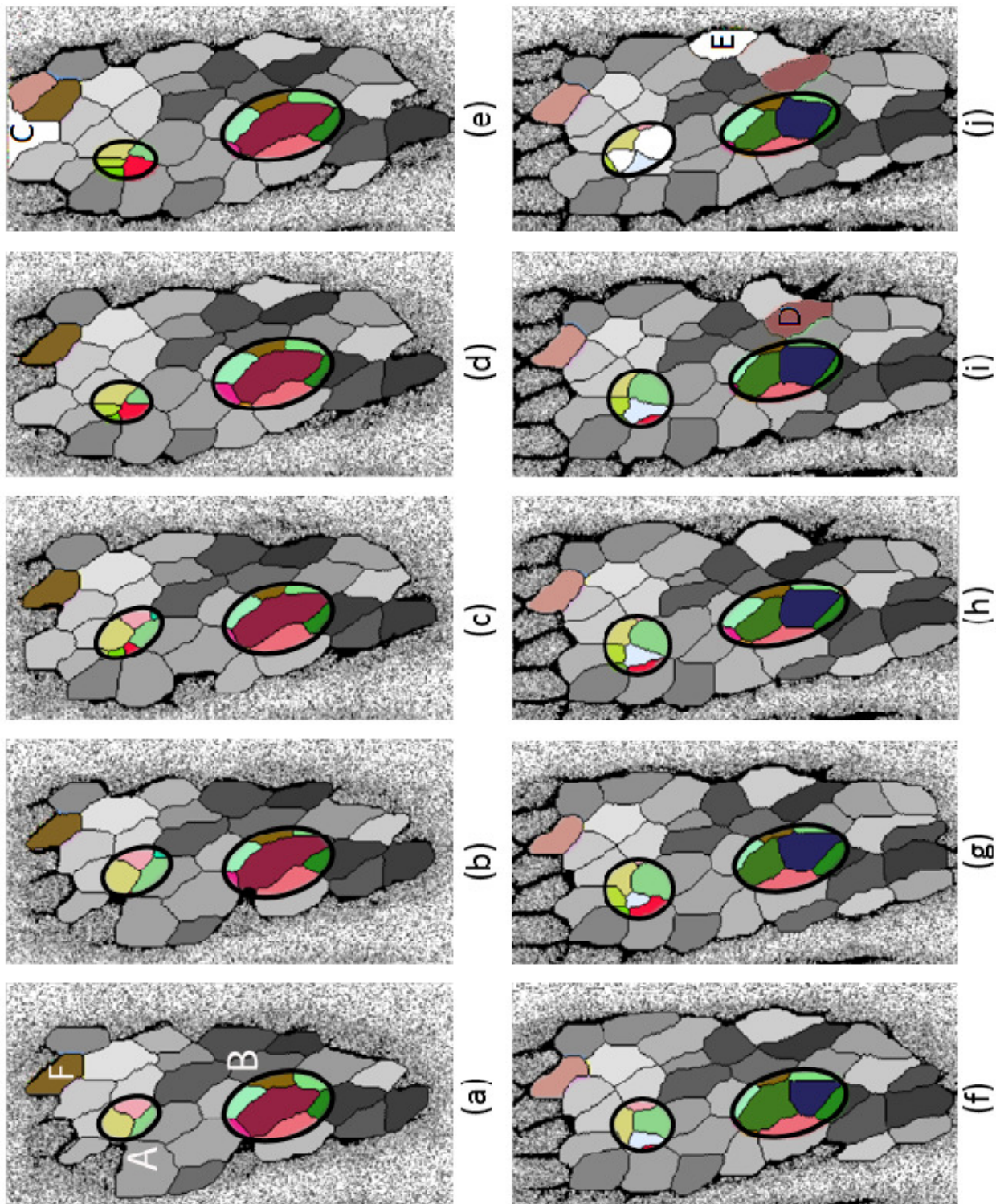


Figure 5.11: CGA errors due to artifacts

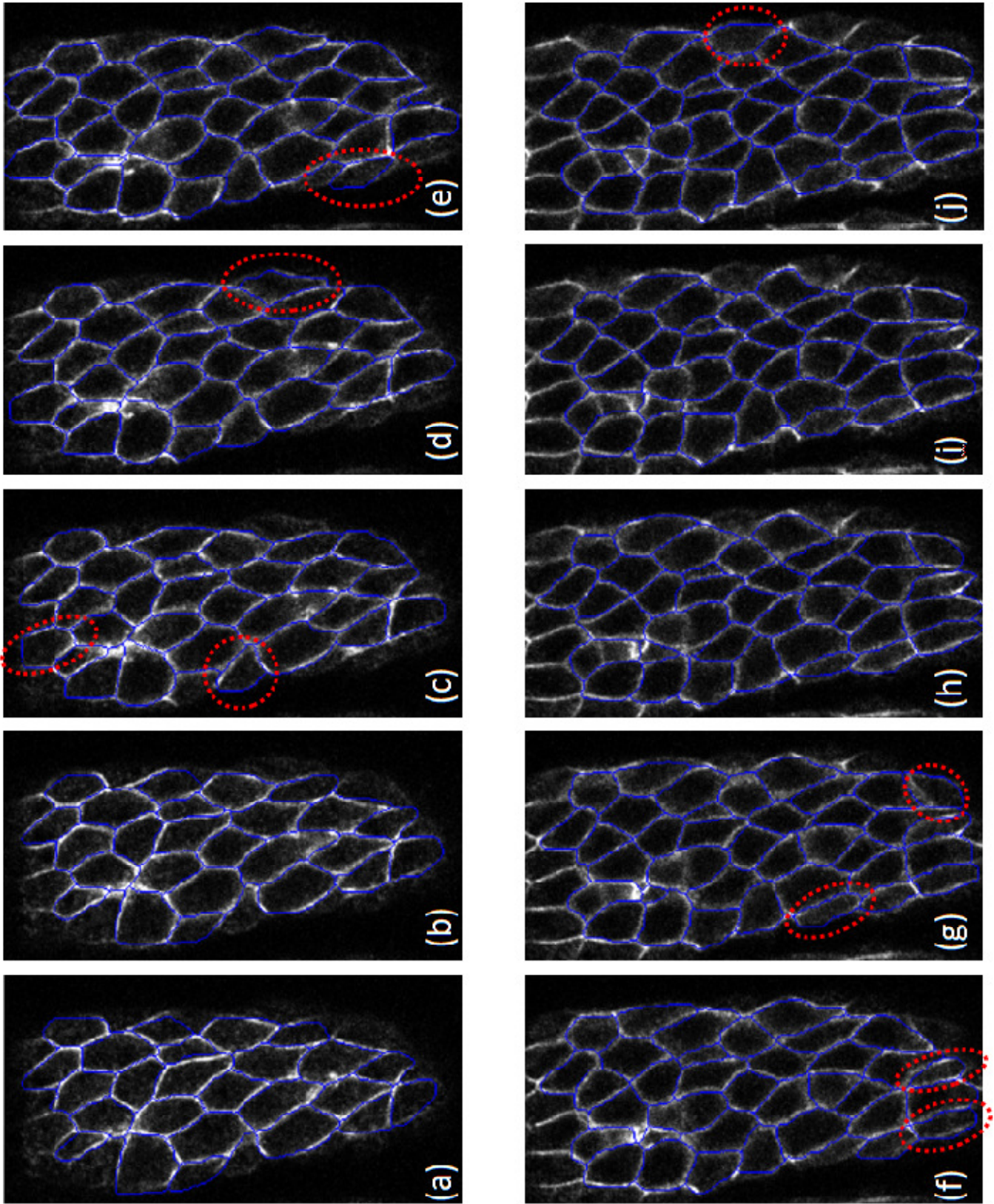


Figure 5.12: Fly-wing edge tracings

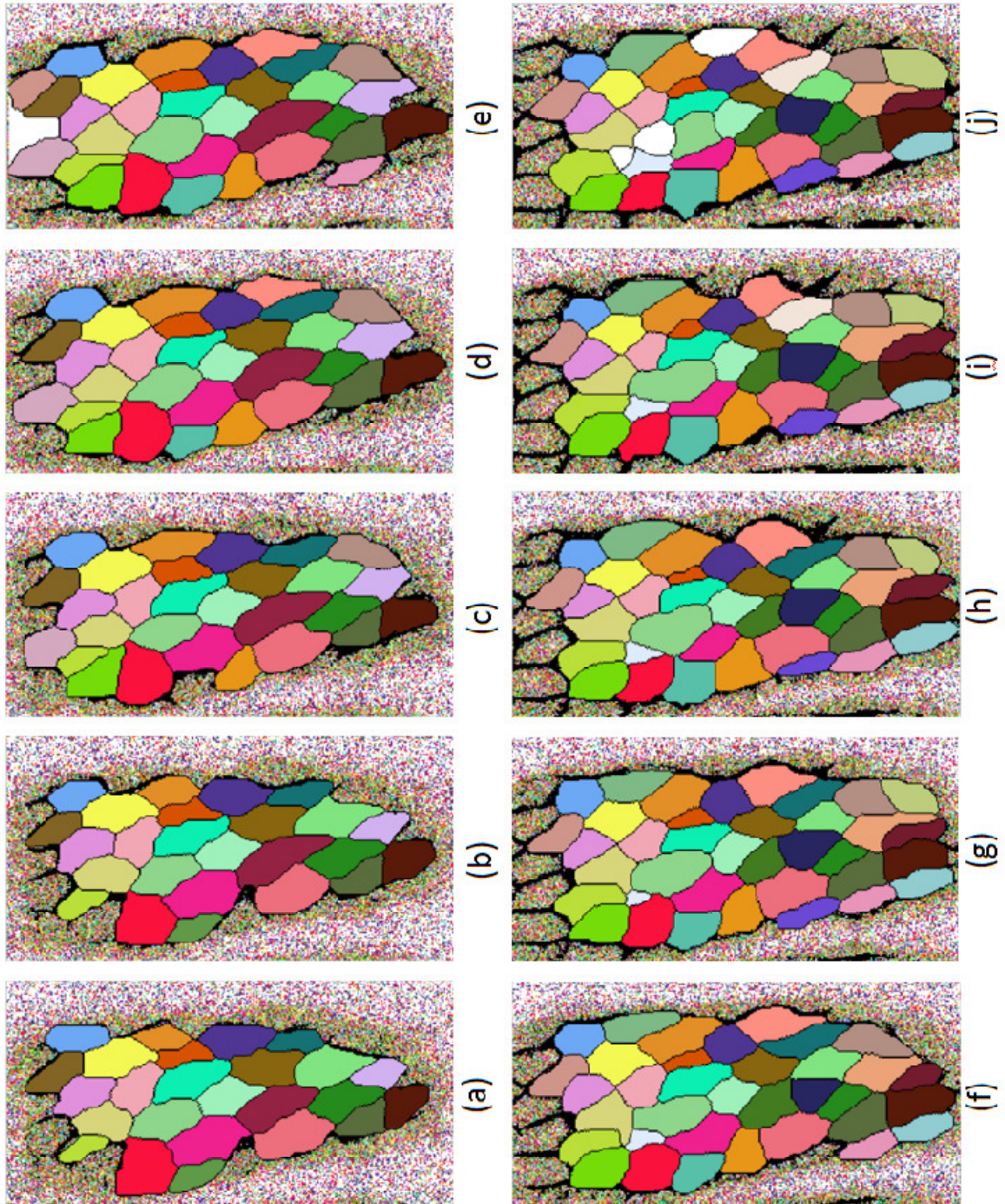


Figure 5.13: CGA results

Additional regions (C, D, E and F) are highlighted in Figure 5.11. C and E are unassigned and have not been grouped into a particular cell. Highlight C is an image boundary artifact (i.e. image boundaries are taken as boundary pixels) while highlight E corresponds to the appearance of a region on the last slice of the stack. This cannot be grouped since no further slices are available for comparison. Highlight D indicates a Type 1 error while highlight F shows a different type of error. Notice the region in Figure 5.11 (e) beside label C. This region is not detected in any slice other than (e) and exists due to the artifact pointed out in C. The CGA has associated this region with those correctly belonging to F. The correct outlines of F are shown in Figure 5.14(b).

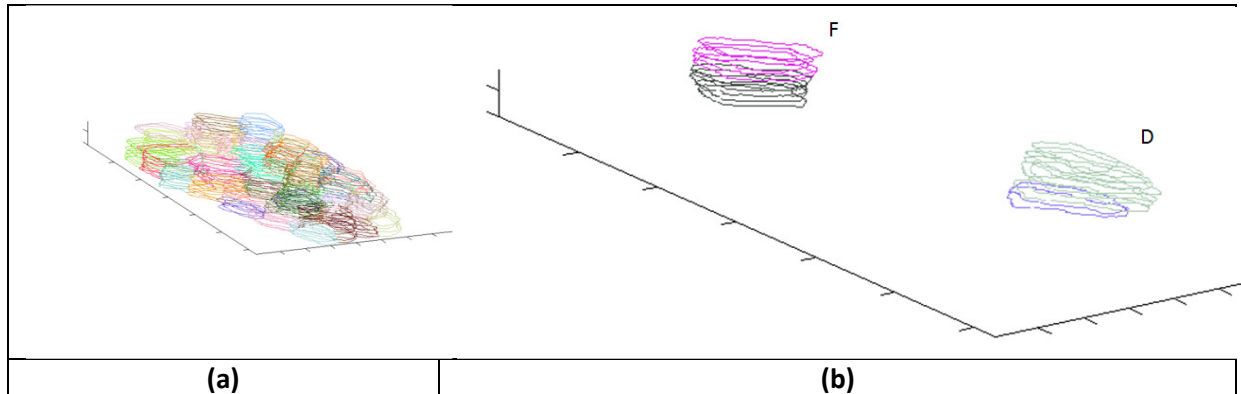


Figure 5.14: CGA results and error resolution

Evaluation

Table 5.18: Fly-wing algorithm statistics

Algorithm Statistics			
Actual Number of Groupings:	41	Number of 2D Contours:	327
Number of Groupings Found:	42	Number of Probabilistic Iterations:	2
Number of Correct Groupings:	39	Number of Type 1 Optimization Iterations:	2
Number of Associations:	280	Number of Type 2 Optimization Iterations:	3
Correct Associations:	278	Run Time:	13.89s

Table 5.19: Fly-wing triplet statistics

Probability Accepted Triplets	Topology Accepted Triplets	Total Number of Triplets
11	10	4192

Table 5.20: Fly-wing errors

Number of Type 1 Errors	Number of Type 2 Errors	Number of Type 3 Errors	Number of Unassigned Regions	Total Number of Errors
1	0	1	4	6

The CGA has produced accurate cell groupings despite various image and edge detection artifacts. The process required 13.89 seconds. Ignoring image artifacts, manual associations indicate forty-one distinct cells. Comparison of manual associations also reveals that the CGA has correctly associated 39 total groupings. This trial has resulted in a grouping accuracy of 90.7%, indicating that the CGA is suitable for use with real confocal images. Notice also that association accuracy is 99.3% and overall accuracy 90.1%. Clearly, the CGA has been successful in associating contours.

Table 5.21: Fly-wing evaluations

Grouping-Cell Match	Correct Groupings	Grouping Accuracy	Association Accuracy	Overall Accuracy
97.6%	92.9%	90.7%	99.3%	90.1%

Nonetheless, CGA errors have been identified as regions D and F (Figure. 5.11). These are similar to those encountered for the synthetic cases, and are potentially corrected with changes in parameter values. A CGA trial with modified parameters is provided in Section 5.4.2.

5.4.2 Fly-wing Z- Series- Modified Parameter Value

Table 5.22: Fly-wing parameter change

Adjusted Parameter	Affects	Original Value	Value
X-Y Triplet Threshold	Probabilistic grouping	85%	80%

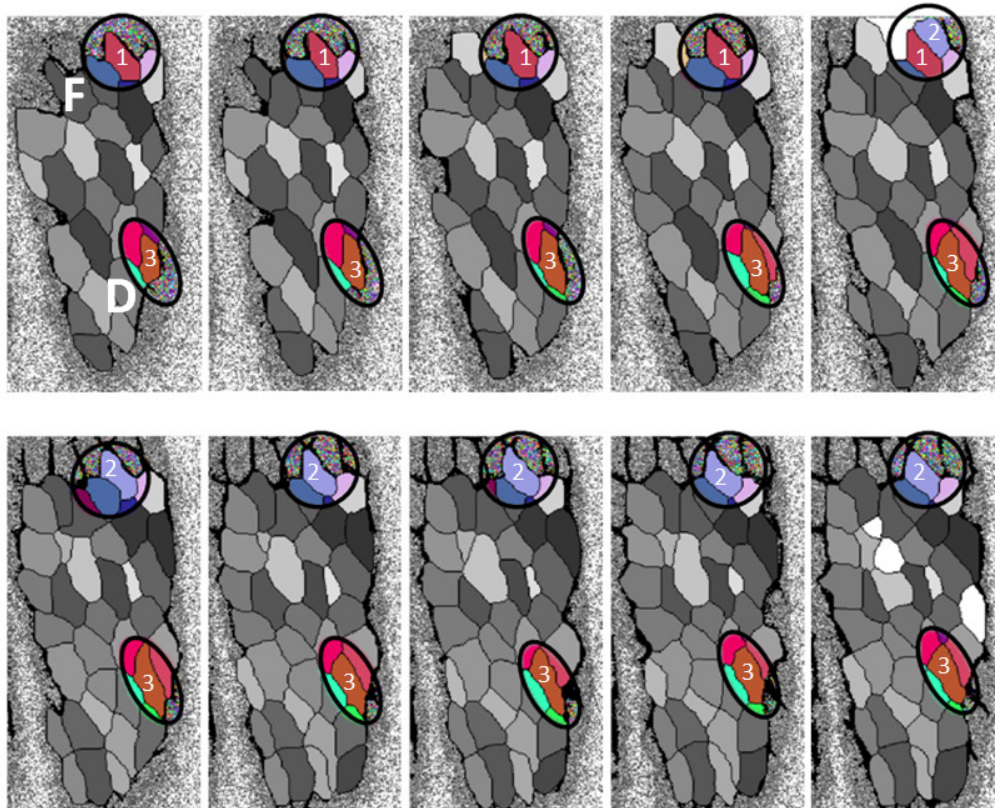


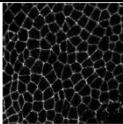
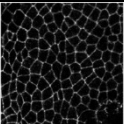
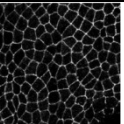
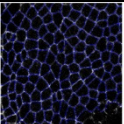
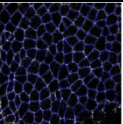
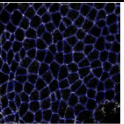
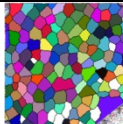
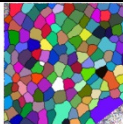
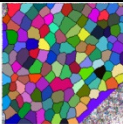
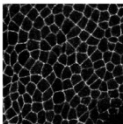
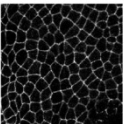
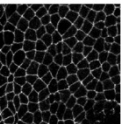
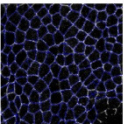
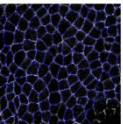
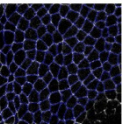
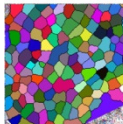
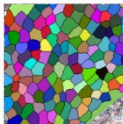
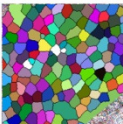
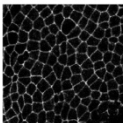
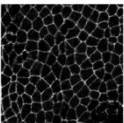
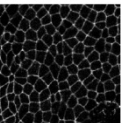
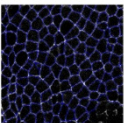
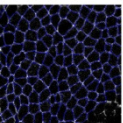
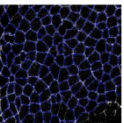
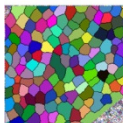


Figure 5.15: Fly-wing parameter change results

Figure 5.15 illustrates the final output with CGA X-Y triplet threshold adjusted from 85% to 80%. The raw images are identical to those used in the initial trial. As seen above, grouping error D has been corrected. All other groupings remain the same and error F still exists. Since error F is caused by inconsistent edge detection, the adjustment of parameters cannot be expected to compensate. Nonetheless, it is seen that adjustment of CGA parameters improves results. Specific combinations of parameter values may be suitable for different image sets, and are not explored here. Suffice it to say, the customizability of CGA parameters is beneficial to the overall performance of the algorithm.

5.4.3 Drosophila Wound Time Series

Table 5.23 summarizes the results obtained from a time-series of confocal image data. This data set serves to show that the CGA is suitable for associating two-dimensional cell tracings across time. That is, the third-dimension need not correspond to the physical z- direction. Visual inspection of Figure 5.16 indicates that the CGA has performed adequately. Unlike previous sections, accuracy statistics are not provided. Rather, a discussion on the effects of edge detection follows.

Table 5.23: Drosophila wound overview

Raw Image			Computer Traced Outlines			Colourized Cell Groupings		
								
								
								

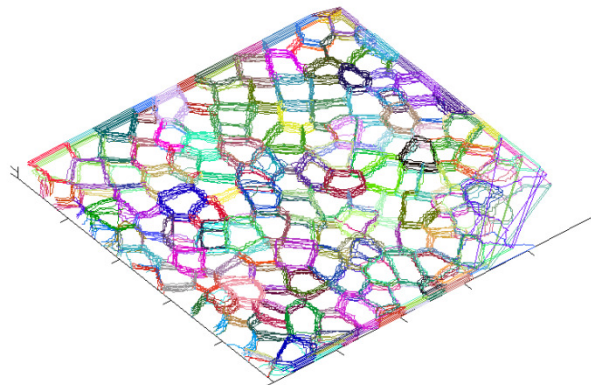
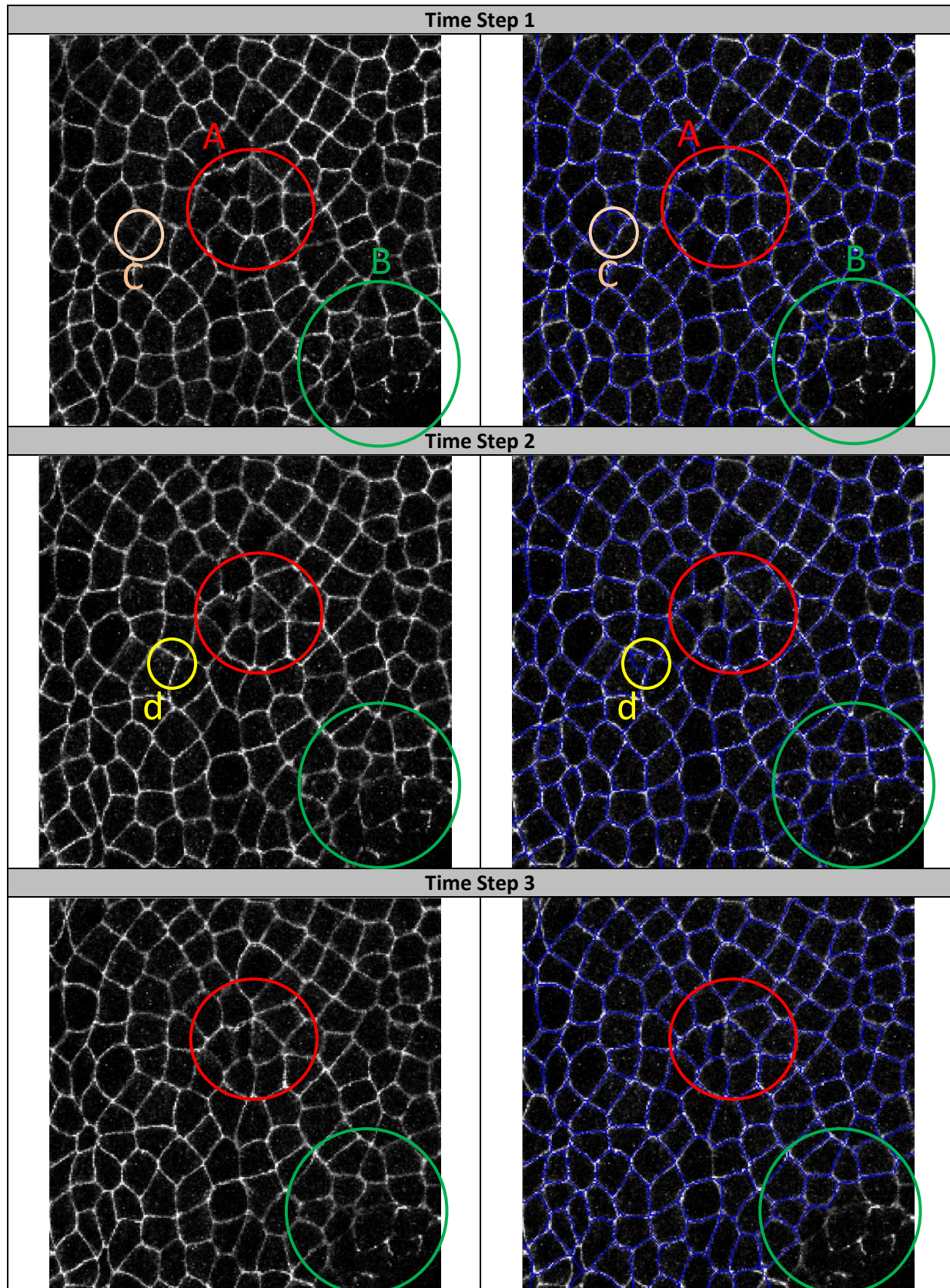


Figure 5.16: Drosophila wound results

Results

The analyzed images have a resolution of 250 x 250 x 9 (rows x columns x time steps). Each time step was separated by 3 minutes. Figure 5.24 depicts enlarged images for time steps 1 through 3. Note the four highlighted areas. These have been highlighted due to faulty edge detection. Across time steps, detection of edges in area A is distorted by fluctuations in image quality (possibly caused by some biological process and/or photo-bleaching). Area B is adjacent to a laser-induced wound, containing cell membranes that have been punctured. In some time steps, these incomplete boundaries are detected by edge detection software. Areas C and D correspond to instances of local noise variation.

Table 5. 24: Drosophila wound image artifacts and edge tracings



For cases A through D, specific image qualities have contributed to undesirable edge detection results. First, areas A and B represent image regions where biologically significant events may be occurring. Distortion of image quality due to such events has caused inconsistent detection of edges. Additionally, the possibility of photo-bleaching due to elapsed time may also contribute to this. Areas C and D are examples of strong local noise, contributing to detection of false edges. In this 250 x 250 pixel image over 3 time steps, four identifiable sources of error exist. Extrapolating to larger images and longer time durations, the potential exists for many more sources of error.

Regardless, the CGA was allowed to run to completion. Figure 5.17 shows the results of the first three time steps. Areas highlighted previously are shown in Figure 5.18. The CGA performed erratically in areas with edge errors but was accurate in others. In a time-series data set, in which cells may propagate, the CGA is still capable of establishing accurate groupings.

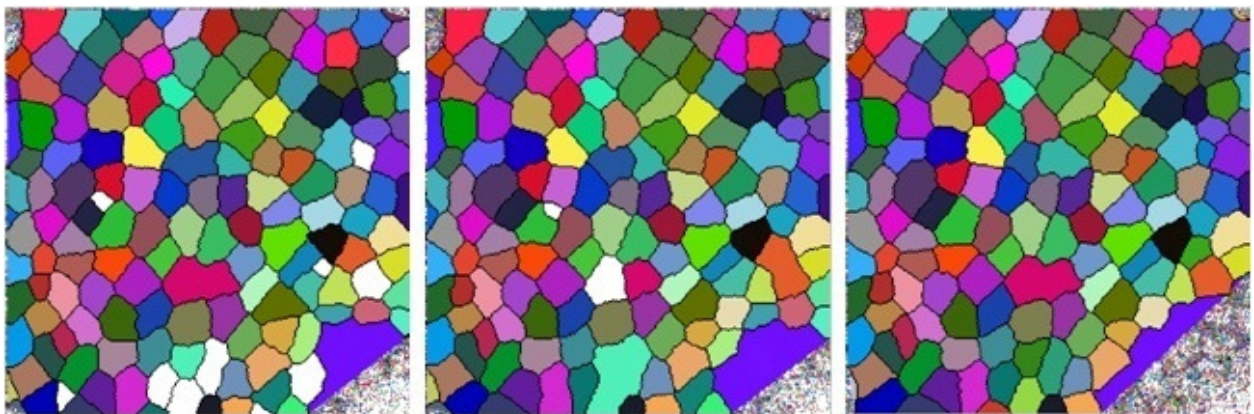


Figure 5.17: Drosophila wound results -- first 3 time steps

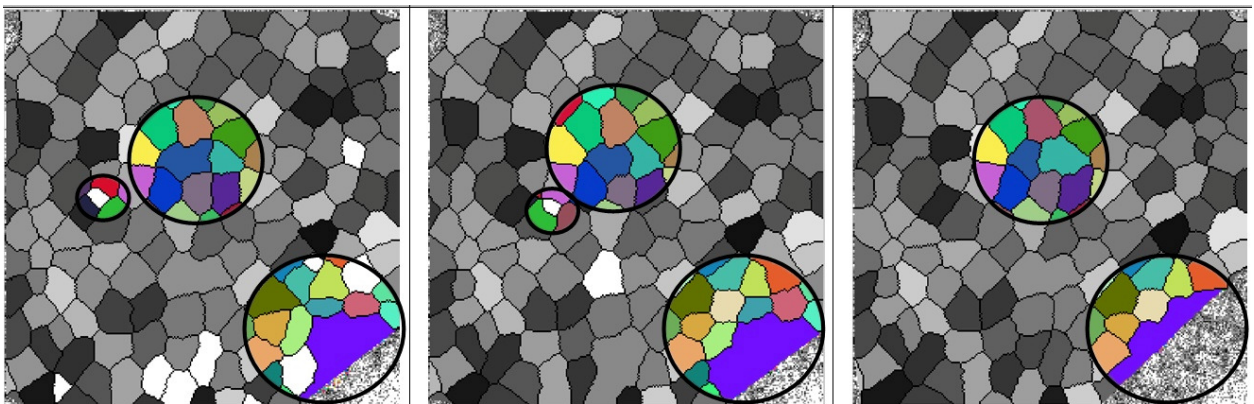


Figure 5.18: Drosophila wound results errors -- first 3 time steps

5.4.4 Observations

The CGA performs well when applied to real confocal image sets. The data can be z-stack or time series images. As described in Figure 5.9, the CGA is run subsequent to edge detections. In itself, the

CGA is strong, but depends upon accurate edge detections. Throughout Section 5.4, the negative effects of faulty edge detection have been described.

As such, these trials imply that edge detection needs to be improved. Particularly, new methods for detecting cell membranes in fluorescent confocal microscopy need to be developed. For these trials, an edge detection algorithm utilizing gradient information and hysteresis thresholding (Lohmann, 1998) was applied. This method is based strongly on well-studied edge detection techniques, and provides good outlines (Figures 5.12 and Table 5.24). However, slight errors in edge detection lead to errors in the CGA, decreasing its effectiveness. The future development of the CGA may include improvements in edge detection as well as reduction of CGA sensitivity to faulty edge tracings.

5.5 Summary

From the various trials presented in this chapter, a number of consistent observations can be drawn about the performance of the CGA. In particular, all trials exceeded 96% association accuracy, suggesting that the CGA is extremely effective in arriving at correct contour associations. However, it was seen that this micro-level accuracy did not translate into macro-level grouping accuracy. It is clear that establishment of correct groupings is very sensitive to errors in association.

Nonetheless, the trials have proven that the CGA framework is robust. Given the accuracies attained for both the cell sheet and cell aggregate, it can be said that the metrics and methodology described in Chapter 3 are strong. Furthermore, the adjustment of various parameters showed that the CGA, through different pathways, is able to arrive at similar groupings. This is particularly encouraging, as it implies that metrics are well-defined, and that customizations to parameter values may be fine-tuned without severe detrimental effects to overall accuracy.

It is noted however that over-constraining of parameters potentially leads to poor CGA groupings. It is therefore critical for users to understand the consequences of parameter adjustment, and to ensure thresholds are not set too strictly. With this in mind, it has also been shown that image properties may strongly affect the effectiveness of the CGA. In the case of slicing distance, it was seen that association accuracy declined as the number of slices decreases. Although grouping accuracy did not necessarily mirror this trend, a decline in association accuracy is alarming.

With respect to real image data, there are a number of factors that play a role in CGA accuracy. Factors such as slicing distance, image protocols and noise, biological occurrences, etc. may all introduce errors to the CGA. Although the CGA has proven strong when analyzing near-ideal data, artifacts pointed out throughout Section 5.4 indicate that improvements in image quality/processing would ensure accurate CGA performance.

Chapter 6: Conclusions and Future Work

The conclusions and future work relate to three categories: 1) image properties and quality, 2) metrics and parameters and 3) algorithm performance. The contour grouping algorithm presented herein represents a major step forward, but further work is required. The progress made here shows that the goal of reconstructing three-dimensional biological cells is attainable.

6.1 Image Properties and Quality

The CGA is more effective with image sets containing higher numbers of slices. Decreasing the number of slices (increasing slicing distance) results in degradation of association accuracy, as well as unexpected algorithm behavior. To this end, it is preferable to maintain image sets in which groupings are expected to have more than 3 contours per cell. Furthermore, the CGA performs well for both z-stacks and time-series images. However, it is sensitive to variations in image properties and quality. The presence of noise and artifacts causes errors to arise. In its present form, the current CGA is seen to be very sensitive to changes/errors in source data.

Since three-dimensional reconstructions rely heavily on image properties and quality, it is recommended that future work also focus upon better understanding of the various image factors affecting CGA performance. From the standpoint of image acquisition, optimal slicing distance, magnification and resolution parameters may exist and a thorough study on this is merited. Additionally, given that the source data are acknowledged to be inherently noisy, improved image processing may also be explored. Such studies may focus upon improved edge detection. Ultimately, the goal of such studies will be to provide a better understanding of images prior to use with the CGA.

6.2 Metrics and Parameters

The metrics and methodology defined in Chapter 3 produce good discrimination characteristics and the algorithm is robust since parameter changes do not significantly affect association/grouping results. Parameter changes could potentially lead to improved results, and certain parameter sets might be potentially more suitable for specific cases (i.e. cell-sheets vs. aggregates). In certain cases, parameters may be inter-related.

It is recommended that future work include a complete parametric study and metric sensitivity analysis. That is, extensive trials on the effects of varying parameters individually, or in groups, is recommended. This may lead to an understanding of parameter values that work well with certain types of data (e.g. cell-sheet or aggregate). Additionally, a metric sensitivity analysis will indicate which metrics play a larger role in CGA decisions. This may result in modification or removal of metrics from the CGA.

6.3 Algorithm Performance and Efficiency

Performance statistics (Chapter 5) imply that the CGA is highly effective in establishing correct contour associations (i.e. > 96% association accuracy). However, having a minimal number of association errors does not guarantee high grouping accuracy. CGA trials demonstrated that correct groupings are highly dependent on near-perfect individual associations. Nonetheless, the CGA shows promise in associating both curved cell-sheets (epithelia) and cell aggregates (sorting cells). Efficiency is largely dependent on the number of accepted triplets, while over-reliance on the cost portion of the CGA leads to poor performance.

It is clear that current CGA performance could be improved. Given that association accuracy is high, the chief consideration is the improvement of overall grouping accuracy. It is recommended that studies on the effects of modifying current CGA methodology be undertaken. For instance, modifications to the cost optimization function, or utilization of other optimizations may be explored (e.g. simulated annealing). Furthermore, fuzzy-logic considerations may be included in analyses. Namely, the probabilistic portion of the CGA may be used to define fuzzy group assignments. Potentially, the flexibility of fuzzy assignments may allow the CGA to correct grouping errors. Similarly, global constraints defining contour group statistics may be introduced. For instance, constraints limiting the maximum volume occupied by a contour group may be useful. Such constraints may aid in automated detection of contour groups in error, providing a possible basis for correcting them. It is hoped that such studies will decrease the number of incorrect groupings.

References

"A to Z of tinnitus." British Tinnitus Association. 2009.

<<http://www.tinnitus.org.uk/index.php?q=node/41>>.

Al-Kofahi, K. A., Can, A., Lasek, S., Szarowski, D. H., Dowell-Mesfin, N., Shain, W., et al.. (2003). Median-based robust algorithms for tracing neurons from noisy confocal microscope images. *Information Technology in Biomedicine, IEEE Transactions on*, 7(4), 302-317.

Amenta, N., Choi, S., Dey, T. K., & Leekha, N. (2000). *A simple algorithm for homeomorphic surface reconstruction*. Clear Water Bay, Kowloon, Hong Kong: ACM.

Amenta, N., Bern, M., & Kamvysselis, M. (1998). *A new voronoi-based surface reconstruction algorithm* ACM.

Badin, P., Bailly, G., Revéret, L., Baciú, M., Segebarth, C., & Savariaux, C. Three-dimensional linear articulatory modelling of tongue, lips and face, based on MRI and video images. *Journal of Phonetics*, 30(3), 533-553.

Barbier De Reuille, P., Bohn-Courseau, I., Ljung, K., Morin, H., Carraro, N., Godin, C., et al.. (2006). Computer simulations reveal novel properties of the cell-cell signaling network at the shoot apex in *Arabidopsis*. *Proceedings of the National Academy of Sciences of the United States of America*, 103(5), p. 1627-1632. doi:10.1073/pnas.0510130103

Barillot, C., Gibaud, B., Scarabin, J. -, & Coatrieux, J. -. (1985). 3D reconstruction of cerebral blood vessels. *Computer Graphics and Applications, IEEE*, 5(12), 13-19.

Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C., udio, & Taubin, G. (1999). The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4), 349-359.

Boissonnat, J., Cazals, F., d\&\#233, & ric. (2000). *Smooth surface reconstruction via natural neighbour interpolation of distance functions*. Clear Water Bay, Kowloon, Hong Kong: ACM.

Boman, B. M., Fields, J. Z., Bonham-Carter, O., & Runquist, O. A. (2001). Computer modelling implicates stem cell overproduction in colon cancer initiation. *Cancer Research*, 61(23), 8408-8411.

Brakenhoff, G. J., van, d. V., van Spronsen, E. A., Linnemans, W. A. M., & Nanninga, N. (1985). Three-dimensional chromatin distribution in neuroblastoma nuclei shown by confocal scanning laser microscopy. *Nature*, 317(6039), 748-749.

Brodland, G. W., Viens, D., & Veldhuis, J. H. (2007). A new cell-based FE model for the mechanics of embryonic epithelia. *Computer Methods in Biomechanics and Biomedical Engineering*, 10(2), 121-128.

- Brodland, G. W. (2002). The differential interfacial tension hypothesis (DITH): A comprehensive theory for the self-rearrangement of embryonic cells and tissues. *Journal of Biomechanical Engineering*, 124(2), 188-197.
- Brodland, G. W., & Veldhuis, J. H. (2002). Computer simulations of mitosis and interdependencies between mitosis orientation, cell shape and epithelia reshaping. *Journal of Biomechanics*, 35(5), 673-681.
- Brooks, R. A., & Chiro, G. D. (1976). Principles of computer assisted tomography (CAT) in radiographic and radioisotopic imaging. *Physics in Medicine and Biology*, 21(5), 689-732.
- Brown, M., Semelka, R., & Nishino, T. K. (2004). MRI: Basic principles and applications, 3rd edition. *Medical Physics*, 31(1), 170.
- Chabrierie, A., Ozlen, F., Nakajima, S., Leventon, M., Atsumi, H., Grimson, E., et al.. (1998). Three-dimensional reconstruction and surgical navigation in pediatric epilepsy surgery. *Medical Image Computing and Computer-Assisted Intervention – MICCAI'98*, 1496, 74-83.
- Chen, H. H., & Brodland, G. W. (2000). Cell-level finite element studies of viscous cells in planar aggregates. *Journal of Biomechanical Engineering*, 122(4), 394-401.
- Chen, X., & Brodland, G. W. (2008). Multi-scale finite element modelling allows the mechanics of amphibian neurulation to be elucidated. *Physical Biology*, 5(1), 015003.
- Cline, H. E., Dumoulin, C. L., Hart, H. R., Jr, Lorensen, W. E., & Ludke, S. (1987). 3D reconstruction of the brain from magnetic resonance images using a connectivity algorithm. *Magnetic Resonance Imaging*, 5(5), 345-352.
- Collins, D. L., Holmes, C. J., Peters, T. M., & Evans, A. C. (1995). Automatic 3-D model-based neuroanatomical segmentation. *Human Brain Mapping*, 3(3), 190-208.
- Dima, A., Scholz, M., & Obermayer, K. (2002). Automatic segmentation and skeletonization of neurons from confocal microscopy images based on the 3-D wavelet transform. *Image Processing, IEEE Transactions on*, 11(7), 790-801.
- Durbin, R., Szeliski, R., & Yuille, A. (1989). An analysis of the elastic net approach to the traveling salesman problem. *Neural Computation*, 1(3), 348-358.
- Fisher, E., Cothren, J., Robert M., Tkach, J. A., Masaryk, T. J., & Cornhill, J. F. (1997). Knowledge-based 3D segmentation of the brain in MR images for quantitative multiple sclerosis lesion tracking. , 3034(1) 19-25.
- Girod, B., Greiner, G., & Niemann, H. (2000). *Principles of 3D image analysis and synthesis*. Boston: Kluwer Academic Publishers.

- Groh, P. G. (2008). *Multi-view imaging of drosophila embryos*. Waterloo, Ont.: University of Waterloo.
- Gull, S. F., & Newton, T. J. (1986). Maximum entropy tomography. *Applied Optics*, 25(1), 156-160.
- Heim, R., & Tsien, R. Y. Engineering green fluorescent protein for improved brightness, longer wavelengths and fluorescence resonance energy transfer. *Current Biology*, 6(2), 178-182.
- Hendrickson, B., & Kolda, T. G. (2000). Graph partitioning models for parallel computing. *Parallel Computing*, 26(12), 1519-1534.
- Hutson, M. S., Brodland, G. W., Yang, J., & Viens, D. (2008). Cell sorting in three dimensions: Topology, fluctuations, and fluidlike instabilities. *Physical Review Letters*, 101(14), 148105.
- Iniewski, K. (2009). *Medical imaging : Principles, detectors, and electronics*. Hoboken, N.J.: Wiley.
- Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A review. *ACM Comput.Surv.*, 31(3), 264-323.
- Jones, T., & Metaxas, D. (1997). Automated 3D segmentation using deformable models and fuzzy affinity. *Information Processing in Medical Imaging*, 1230, 113-126.
- Kaufman, S. C., Musch, D. C., Belin, M. W., Cohen, E. J., Meisler, D. M., Reinhart, W. J., et al.. (2004). Confocal microscopy: A report by the american academy of ophthalmology. *Ophthalmology*, 111(2), 396-406.
- Kwon, Y. H., Wells, K. S., & Hoch, H. C. (1993). Fluorescence confocal microscopy: Applications in fungal cytology. *Mycologia*, 85(5), 721-733.
- Laissue, P. P., Reiter, C., Hiesinger, P. R., Halter, S., Fischbach, K. F., & Stocker, R. F. (1999). Three-dimensional reconstruction of the antennal lobe in *drosophila melanogaster*. *The Journal of Comparative Neurology*, 405(4), 543-552.
- Larrañaga, P., Kuijpers, C. M. H., Murga, R. H., Inza, I., & Dizdarevic, S. (1999). Genetic algorithms for the travelling salesman problem: A review of representations and operators. *Artificial Intelligence Review*, 13(2), 129-170.
- Levman, J., Leung, T., Causer, P., Plewes, D., & Martel, A. L. (2008). Classification of dynamic contrast-enhanced magnetic resonance breast lesions by support vector machines. *Medical Imaging, IEEE Transactions on*, 27(5), 688-696.
- Lin, G., Chawla, M. K., Olson, K., Barnes, C. A., Guzowski, J. F., Bjornsson, C., et al.. (2007). A multi-model approach to simultaneous segmentation and classification of heterogeneous populations of cell nuclei in 3D confocal microscope images. *Cytometry Part A*, 71A(9), 724-736.

- Lohmann, G. (1998). *Volumetric image analysis*. Chichester, West Essex; New York, N.Y.: Stuttgart: Wiley; Teubner.
- Longo, D., Peirce, S. M., Skalak, T. C., Davidson, L., Marsden, M., Dzamba, B., et al.. (2004). Multicellular computer simulation of morphogenesis: Blastocoel roof thinning and matrix assembly in *xenopus laevis*. *Developmental Biology*, 271(1), 210-222.
- Magoulas, G., & Prentza, A. (2001). Machine learning in medical applications. *Machine Learning and Its Applications*, 2049, 300-307.
- Malek, M., Guruswamy, M., Pandya, M., & Owens, H. (1989). Serial and parallel simulated annealing and tabu search algorithms for the traveling salesman problem. *Annals of Operations Research*, 21(1), 59-84.
- Milligan, G. W., & Cooper, M. C. (1987). Methodology review: Clustering methods. *Applied Psychological Measurement*, 11(4), 329-354.
- Nielson, G. M. (2003). On marching cubes. *Visualization and Computer Graphics, IEEE Transactions on*, 9(3), 283-297.
- Paulus, M. J., Gleason, S. S., Kennel, S. J., Hunsicker, P. R., & Johnson, D. K. (2000). High resolution X-ray computed tomography: An emerging tool for small animal cancer research. *Neoplasia (New York, N.Y.)*, 2(1-2), 62-70.
- Pawley, J. B. (2006). *Handbook of biological confocal microscopy* (3rd ed ed.). New York, NY: Springer.
- Peterson, C. (1990). Parallel distributed approaches to combinatorial optimization: Benchmark studies on traveling salesman problem. *Neural Computation*, 2(3), 261-269.
- Ramm, A. G. (. G.), & Katsevich, A. I. (. I. (1996). *Radon transform and local tomography*. Boca Raton: CRC Press.
- Sarti, A., Ortiz de Solorzano, C., Lockett, S., & Malladi, R. (2000). A geometric model for 3-D confocal image analysis. *Biomedical Engineering, IEEE Transactions on*, 47(12), 1600-1609.
- Schierlitz, L., Dumanli, H., Robinson, J. N., Burrows, P. E., Schreyer, A. G., Kikinis, R., et al.. (2001). Three-dimensional magnetic resonance imaging of fetal brains. *The Lancet*, 357(9263), 1177-1178.
- Solórzano, C., Rodriguez, E., Jones, A., Pinkel, D., Gray, J., Sudar, D., et al.. (1999). Segmentation of confocal microscope images of cell nuclei in thick tissue sections. *Journal of Microscopy*, 193(3), 212-226.
- Stout, S. D., Brunsdon, B. S., Hildebolt, C. F., Commean, P. K., Smith, K. E., & Tappen, N. C. (1999). Computer-assisted 3D reconstruction of serial sections of cortical bone to determine the 3D structure of osteons. *Calcified Tissue International*, 65(4), 280-284.

Udupa, J. K. (1982). Interactive segmentation and boundary surface formation for 3-D digital images. *Computer Graphics and Image Processing*, 18(3), 213-235.

Viens, D., & Brodland, G. W. (2007). A three-dimensional finite element model for the mechanics of cell-cell interactions. *Journal of Biomechanical Engineering*, 129(5), 651-657.

Wang, D., Hassan, O., Morgan, K., & Weatherill, N. (2006). Efficient surface reconstruction from contours based on two-dimensional delaunay triangulation. *International Journal for Numerical Methods in Engineering*, 65(5), 734-751.

Wu, T., Stewart, A., Stanton, M., McCauley, T., Phillips, W., Kopans, D. B., et al.. (2003). Tomographic mammography using a limited number of low-dose cone-beam projection images. *Medical Physics*, 30(3), 365-380.

Yang, T. (2008). *Finite element studies of an embryonic cell aggregate under parallel plate compression*. Waterloo, Ont.: University of Waterloo.

Zha, H., He, X., Ding, C., Simon, H., & Gu, M. (2001). *Bipartite graph partitioning and data clustering*. Atlanta, Georgia, USA: ACM.